

# 概要

近年、機械翻訳の方式として等価的類推思考の原理に基づく機械翻訳方式が提案されている。この方式の実現に向けて、日本語の重文・複文を対象とした文型パターンを大量に蓄積した文型パターン辞書の構築が進められている。文型パターンは、意味的等価性を保ち訳出に有効であるために、文の構造を支える単語は「字面」「変数」あるいは「関数」に置き換えられる。また、そうでないものは、文型パターンの構成要素にならない。従来、入力文の解析は、入力文と文型パターンを照合して、入力文に適合する文型パターンを抽出する方式で行なわれている。この入力文を解析する方式に対する課題は、「できるだけ多くの文が文型パターンに適合すること」、すなわち「文型パターンの網羅性を確保すること」である。文型パターンの網羅性を確保するには、文型パターンを大量に作成することと、文型パターンの汎用性を高めること（汎化）が考えられる。一般のパターン型知識ベースを開発する上で、パターン数を増やすコストや、汎用性の高いパターンの記述を行うコスト（汎化作業コスト）は、実質的に重要なファクタであるため、知識ベースの開発事例における実測値に対する関心は高い。

現在、重文・複文を対象とした文型パターン辞書の開発事例において、文型パターンの汎用性を高めるために、「記号」「関数」が文型パターンに記述されている。しかし、「記号」と「関数」を記述することにより、文型パターン汎化効果がどの程度あるのか不明であり、汎化作業コストとの関係も不明である。そこで本稿では、「記号」と「関数」による文型パターンの「汎化効果の評価」を行ない、「記号」と「関数」の有用性を確認する。また、「記号」と「関数」を記述する際にかかる「汎化作業コスト」については、文型パターン辞書の開発事例から調査する。そして、「汎化効果の評価」、および「汎化作業コスト」から、「記号」と「関数」による文型パターン汎化効果の「総合評価」を行なう。

文型パターンに記述されている記号は、文型パターンに表記・表現の揺らぎの指定、省略可能な主語・目的語の指定、任意の要素の指定、語順と位置変更可能要素の指定ができる。関数は、文型パターンに要素の語形指定、付属語類の指定、関数のグループ化の指定ができる。

汎化効果の評価に用いる評価パラメータとしては、「文型再現率  $R1$ 」や、「適合文型数  $N$ 」などが提案されているが、その評価パラメータでは、文型パターン数と汎化効果との関係がわからない。そこで本稿では、網羅性の向上に比例した文型パターン数の換算値を算出できる評価パラメータ「文型パターン拡大率  $\eta$ 」を用いる。 $\eta$ を用いる利点は、汎化効果を文型パターン数で確認できる点である。汎化作業コストは、「記号」や「関数」

を付与する際の文型パターン辞書の組織的な開発過程におけるアナリストやプログラマの作業工数をベースに算出する。総合評価は、「汎化効果の評価」と「汎化作業コスト」から、1人日当たりの文型パターン作成数に換算した値を用いて評価する。

全（単語・句・節）レベルの「記号」、「関数」による文型パターンの汎化効果の評価の結果、記号で最も汎化効果があるのは、任意の要素の指定ができる離散記号と分かった。離散記号を文型パターン辞書の記述に用いると、文型パターン辞書を3.23倍に拡大したことが分かった。関数で最も汎化効果があるのは、関数のグループ化の指定ができる同値型グループ関数と分かった。同値型グループ関数を文型パターン辞書の記述に用いると、文型パターン数が1.01倍に拡大したことが分かった。全ての記号・関数を文型パターン辞書の記述に用いると、文型パターン数が3.23倍に拡大したことが分かった。作業コストが最も低いのは、文節境界記号の195人日で、最も高いのは、関数の1,125日だと分かった。全レベルの「記号」、「関数」による文型パターンの汎化効果の評価結果と作業コストから、総合評価が最も良いのは、離散記号で、1人日あたり668パターン作成出来ることに相当することが分かった。

これらから、記号・関数は、文型パターンの汎用性を大幅に向上させることが分かった。将来、一般のパターン型知識ベースを開発する上で、本稿で求めた総合評価を用いれば、パターン数を増やすコストや、汎用性の高いパターンの記述を行うコストの見積りができ、パターンの網羅性を効率よく確保することが可能となると考えられる。

# 目次

<b>1</b>	<b>はじめに</b>	<b>1</b>
<b>2</b>	<b>文型パターンにおける記号・関数</b>	<b>3</b>
2.1	文型パターン辞書	3
2.1.1	文型パターンの概要	3
2.1.2	文型パターンの構成	3
2.1.3	3レベルの定義	4
2.1.4	文型パターンの構成要素	4
2.2	文種別	5
2.3	記号について	6
2.3.1	離散記号	6
2.3.2	文節境界記号	8
2.3.3	要素選択記号	8
2.3.4	任意要素記号	9
2.3.5	順序任意記号	9
2.3.6	位置変更可能記号	10
2.4	関数について	11
2.4.1	日本語様相関数	11
2.4.2	同値型グループ関数	13
<b>3</b>	<b>評価パラメータ</b>	<b>14</b>
3.1	網羅性の評価	14
3.1.1	文型再現率 $R1$	15
3.1.2	適合文型数 $N$	15
3.1.3	単語一致率 $R2\_W$	15
3.1.4	$R1, N, R2\_W$ の求め方	15
3.2	パターンの増加数への換算	17
3.3	文型パターン拡大率 $\eta$	17
<b>4</b>	<b>汎化効果の評価</b>	<b>21</b>
4.1	汎化効果の計測方法	21

4.1.1	基準パターン	21
4.1.2	記号・関数の削除	21
4.1.3	計測方法	24
4.1.4	計測の様子	26
4.2	汎用性の評価結果	31
<b>5</b>	<b>汎化作業コスト</b>	<b>33</b>
5.1	汎化作業の関係	33
5.2	記号・関数に対する汎化作業コスト	34
<b>6</b>	<b>総合評価</b>	<b>35</b>
<b>7</b>	<b>考察</b>	<b>37</b>
7.1	意味的排他性の評価	37
7.1.1	意味的排他性の評価パラメータ	37
7.1.2	排他性の評価結果	37
7.1.3	文型パターン数の換算値と排他性の評価結果	38
7.1.4	離散記号の効果があった例	38
<b>8</b>	<b>おわりに</b>	<b>40</b>

## 表目次

1	3レベルの定義	4
2	必須要素と任意要素の区別	4
3	文種別の説明と例一覧	5
4	使用する記号の機能と記述形式	6
5	使用する関数の機能と記述形式	12
6	評価パラメータの結果（文法・単語レベル）	31
7	評価パラメータの結果（文法・句レベル）	31
8	評価パラメータの結果（文法・節レベル）	32
9	評価パラメータの結果（文法・全レベル）	32
10	基礎となる言語的情報の作業工数	34
11	記述子ごとの導入作業の工数と作業コスト	34
12	総合評価結果（単語レベル）	35
13	総合評価結果（句レベル）	35
14	総合評価結果（節レベル）	36
15	総合評価結果（全レベル）	36

## 目次

1	$R1, N, R2\_W$ の算出方法 . . . . .	16
2	$R1$ と文型パターン数の関係図 (文法・単語レベル) . . . . .	19
3	$R1$ と文型パターン数の関係図 (文法・句レベル) . . . . .	19
4	$R1$ と文型パターン数の関係図 (文法・節レベル) . . . . .	20
5	$R1$ と文型パターン数の関係図 (文法・全レベル) . . . . .	20
6	検索ツールを用いた計測の様子 . . . . .	26

# 1 はじめに

等価的類推思考の原理に基づいて，品質の良い機械翻訳や言い換え技術などを実現するために，日本語の重文・複文を対象とした「文型パターン辞書」の構築が進められている [1]．文型パターンは，日英文対応の対訳コーパスから作成され，言語表現を，「字面」，「変数」，「関数」，「記号」に書き換えた表現で，変数の文法的属性に着目して，「単語レベル」，「句レベル」，「節レベル」の3レベルから構成される．現在，入力文の解析は，入力文と文型パターンを照合し，入力文に適合する文型パターンを抽出する方法で行なっている．この解析に対する課題は，「できるだけ多くの文が文型パターンに適合すること」，すなわち「文型パターンの網羅性を確保すること」である．そのため，文型パターンを大量に作成することと，文型パターンの汎用性を高めること（汎化）が要求される．一般のパターンに基づく知識ベースを開発する上で，パターン数を増やすコストや，汎用性の高いパターンの記述を行うコスト（汎化作業コスト）は，実質的に重要なファクタであるため，知識ベースの開発事例における実測値に対する関心は高い．

現在，重文・複文を対象とした文型パターン辞書の開発事例において，文型パターンの汎用性を高めるために「記号」，「関数」が文型パターンに記述されている [2]．しかし「記号」と「関数」を記述することにより，文型パターン汎化効果がどの程度あるのか不明であり，汎化作業コストとの関係も不明である．そこで本稿では「記号」と「関数」による文型パターンの「汎化効果の評価」を行ない「記号」と「関数」の有用性を確認する．また「記号」と「関数」を記述する際にかかる「汎化作業コスト」については，文型パターン辞書の開発事例から調査する．そして「汎化効果の評価」，および「汎化作業コスト」から「記号」と「関数」による文型パターン汎化効果の「総合評価」を行なう．

汎化効果の評価は，単純に言えば文型パターンの網羅性を評価するパラメータ「文型再現率  $R1$ （入力文に対して適合する文型パターンがある割合）」，「適合文型数  $N$ （入力文に適合した文型パターンの平均）」で評価できるが [3]，その評価パラメータでは，文型パターン数と汎化効果との関係がわからない．そこで本稿では，網羅性の向上に比例した文型パターン数の換算値を算出できる評価パラメータ「文型パターン拡大率  $\eta$ 」を用いる [4]． $\eta$ を用いる利点は，汎化効果を文型パターン数で確認できる点である．「記号」，「関数」による文型パターンの汎化効果の評価を行なう方法は，次の通りである．まず，変数の汎化レベルごとに文型パターンの汎化効果を評価するため，文型パターン辞書（217,703パターン）を「単語レベル（119,229パターン）」，「句レベル（75,708パター

ン)」、「節レベル(22,766パターン)」に分け、各々のレベルごとに「基準パターン辞書」を作成する。そして、各々の基準パターン辞書に対して「記号」、「関数」を削除したパターン辞書を作成する。これを「対象パターン辞書」とする。次に、入力文として対訳標本からランダムに選んだ10,000文と基準パターン辞書、および、対象パターン辞書を検索ツール(SPM)を用いて照合を行なう。そして、照合結果から、評価パラメータを用いて、「記号」、「関数」による文型パターンの汎化効果を定量的に評価する。

汎化作業コストは、文型パターン辞書の組織的な開発過程におけるアナリストやプログラマの作業工数をベースに算出する。具体的な算出方法は、次の通りである。記号・関数は文型パターンに記述する際、日英対訳文対における統語的・意味的な関係をアナリストが分析し、記述可能であるかどうかを検査している。その前後には実務的な作業として、基礎的な作業と記述子固有の作業がある。汎化作業コストは、この基礎的な作業と記述子固有の作業の工数の和で算出する。

総合評価は、「汎化効果の評価」と「汎化作業コスト」から人日あたりの文型パターン作成数に換算した値を用いて評価する。具体的な算出方法は、次の通りである。「汎化効果の評価」で求めた文型パターン数の換算値を「汎化作業コスト」で求めた作業コストで割ることで算出する。

本稿の構成は以下の通りである。第2章で文型パターンにおける記号と関数について述べる。第3章で汎化効果を評価を調査する評価パラメータについて述べる。第4章で「汎化効果の評価」について述べる。第5章で「汎化作業コスト」について述べる。第6章で「総合評価」について述べる。第7章で考察を述べる。第8章でまとめを述べる。



## 2 文型パターンにおける記号・関数

### 2.1 文型パターン辞書

#### 2.1.1 文型パターンの概要

文型パターンとは、日英対訳標本文中の線形な要素を対象に、変数化、関数化、任意化、などを行なったもので、変数化された対象の文法的属性に着目して、単語レベル、句レベル、節レベルの3グループから構成される。ここで、線形要素とは、特定の概念を表現するための表現構造の要素のうち、他の要素に置き換えても表現構造全体の意味が変わらないとき、その要素を表現構造全体の「線形要素」という（それ以外は、「非線形要素」という）。

#### 2.1.2 文型パターンの構成

原文 ( $L$ ) と単語 ( $W$ )、句 ( $P$ )、節 ( $C$ ) の3レベルがあり、各レベルに日 ( $J$ )、英 ( $E$ ) 2つの文型パターンが存在する。以下に原文 ( $LJ, LE$ )、単語 ( $WJ, WE$ )、句 ( $PJ, PE$ )、節 ( $CJ, CE$ ) の文型パターンの例を示す。なお、文型パターンに記述されている「N」や「/y」などの記述子の意味は付録1に示す。

(文型パターンの例)

$LJ$  = 彼は祈るかのように天を仰いだ。

$LE$  = He looked up at the sky as if to pray.

$WJ$  = /y  $\$1^{\wedge}\{tcfk$  N1 は } /cf V2 かのよう $\$1$  /ytck N3 を /cf V4.kako。

$WE$  = N1 V4 $\wedge$ past at N3 as if to V2 $\wedge$ base.

$PJ$  = /y  $\$1^{\wedge}\{tcfk$  N1 は } /cf V2 かのよう $\$1$  /y VP3.kako。

$PE$  = N1 VP3 $\wedge$ past as if to V2 $\wedge$ base.

$CJ$  = /y  $\$1^{\wedge}\{tcfk$  N1 は } /cf V2 かのよう $\$1$  /ytck CL3.kako。

$CE$  = CL3 as if to V2 $\wedge$ base.

### 2.1.3 3レベルの定義

日英対訳原文から文型パターンを作成する際、変数とする領域における日本語の文法的な条件にレベルを設け、単語、句、節の3レベルを作成している。単語、句、節の3レベルの定義を表1に示す。

表 1: 3レベルの定義

汎化レベル	汎化内容
単語レベル	日本語の線形要素とみなせる自立語を変数化した文型パターンのレベル
句レベル	日本語の線形要素とみなせる名詞句、動詞句を変数化すると共に、単語レベルで作成された文型パターンの単語変数を可能な限り句の変数に置き換えた文型パターンのレベル
節レベル	日本語の線形要素とみなせる節を変数化した文型パターンのレベル

### 2.1.4 文型パターンの構成要素

文型パターンは、表2に示す「必須要素」と「任意要素」から構成される。

表 2: 必須要素と任意要素の区別

分類	細分類と説明	
必須要素	日本語文型パターン内にその要素がないと、対応する英語文型パターンが決定できない(すなわち全体が意味をなさない)要素で、字面、変数、関数で表される。	
任意要素	原文任意要素	線形要素のうち、その要素が削除されても英語文型自体は変化せず、英文生成上も何ら問題のないもの。実際の文型パターンの定義では、その位置のみが指定され、どの形式の表現かは指定されない。主に離散記号で表現される。
	文型パターン任意要素	その要素が削除されても英語文型自体は変化しないが、それ自身の訳語の決定や訳語挿入位置の決定が困難であるなど、要素自身の翻訳に困難さが生じるもので、文型パターン定義では「必須要素」と区別して使用される。任意要素記号などで表現される。

文型パターンは日英ともに、「字面」、「変数」、「関数」、「記号」の4種類の記述要素で構成される(付録1)。

- ◆ 「字面」は、日英を対応付けて変域を持たせることができない重文・複文としての意味の固定要素部分、すなわち非線型部分を指定し、それぞれの言語で使用される

文字で記述される。日本語側では、その文型パターンに適合するための字面での照合条件となり、英語側では、その字面の英語を訳出することを指定する訳出条件となる。

- ◆ 「変数」は、自立語のうち日英を対応付けて、定められた値域の範囲で置き換え可能な要素を持つ部分、すなわち線形部分を指定する。変数には変数番号が付いており、日英の対応する変数には同一の変数番号が付与されている。日本語文型パターンの照合に成功した場合は、日本語変数に照合した日本語表現の訳語が、同一の変数番号を持つ英語の変数に橋渡しされ、英語パターンで訳出される仕組みとなっている。変数は、変数名そのものによって文法的な変域が規定され、更に日本語の変数は意味属性によって意味的な変域も規定される。
- ◆ 「関数」は、変数に付属する付属語の指定や、変数または関数に対する文法的な指定を行うもので、表記上の揺らぎを吸収し、適用範囲の広い文型パターンが記述できるようになる。
- ◆ 「記号」は、文型パターン全体の非線型な構造を保ちながら、OR 条件による線形要素の適用範囲の拡大、線形要素の挿入や省略、線形要素の移動など、構造上の線型性を記述するために使用される。

## 2.2 文種別

文種別は日本語原文が接続箇所と埋め込み箇所を何箇所持つかを表していて、接続箇所と埋め込み箇所の数で文種別 1～5 が分かれている。各文種別の説明と例を以下に示す。

表 3: 文種別の説明と例一覧

番号	分類	例文
文種別 1	文接続 1 箇所を持つ文	私は椅子に <span style="border: 1px solid black; padding: 0 2px;">座り</span> , 窓の外を眺めた。
文種別 2	文接続 2 箇所を持つ文	明日は東京へ <span style="border: 1px solid black; padding: 0 2px;">行っ</span> て、友達に <span style="border: 1px solid black; padding: 0 2px;">会い</span> , 久しぶりに話をする。
文種別 3	埋め込み文 1 つを含む文	さっきまで <span style="border: 1px solid black; padding: 0 2px;">ここにいた</span> 猫がいなくなった。
文種別 4	埋め込み文 2 つを含む文	<span style="border: 1px solid black; padding: 0 2px;">父親と撮った</span> 写真を <span style="border: 1px solid black; padding: 0 2px;">アメリカにいる</span> 姉に送った。
文種別 5	文接続 1 箇所 + 埋め込み文 1 つの文	<span style="border: 1px solid black; padding: 0 2px;">彼女の吐き出す</span> 息は <span style="border: 1px solid black; padding: 0 2px;">白く</span> , 頬は赤かった。

は文接続箇所を,  は埋め込み箇所を表す

## 2.3 記号について

本稿で取り扱う記号について説明する。記号は、文型パターン全体の非線型な構造を保ちながら、構造上の線型性を記述するために用いられている。文型パターンに使用されている記号の中から、日本語文型パターンに用いられており、原文と適合するように削除できる6つの記号(9種類)を取り扱う(表4)。以下に、取り扱うそれぞれの記号について説明する。

表 4: 使用する記号の機能と記述形式

記号名	機能	記述形式	記述例とその意味
離散記号	文型パターンにない要素で、挿入可能なものの指定する。	/c 格要素 /k 形容(動)詞連体形・連体詞 /f 副詞(句) /t 連体節 /y 連用節(接続詞を含む)	/y N1 は、/ck N2 を /f V3 . N1 の前に連用節、N2 の前に格要素と形容詞・形容動詞連体形・連体詞、V3 の前に副詞が挿入してもよいことを示す。
文節境界記号	文節の境界を表す。入力日本語とのマッチング条件として利用する。	!	N1 は ! N2 を ! V3.tai . "は"と"N2", "を"と"V3"の間が文節境界であることを示す。
要素選択記号	複数の選択要素を表す。照合では複数記述された要素のうちいずれかにマッチすればよいとする。	#数(パターン記述1   パターン記述2   ...)	#2(とかく   兎角) "とかく"または"兎角"のいずれかにマッチすればよいという条件を示す。
任意要素記号	照合ではその要素が入力日本語にあってもなくてもよいとする。	#数 [パターン記述]	#3[ADV4] ADV4 が照合の際にあってもなくてもよいことを示す。
順序任意記号	照合の際に隣接した要素が入れ替わってもよいとする。	#数(パターン記述1, パターン記述2, ...)	#1{/N2 が, /N3 で} "N2 が"と"N3 で"が順序入れ替え可能であることを示す。
位置変更可能記号	照合の際に要素の移動可能な位置を指定する。	対象要素...\$数 {パターン記述} 移動先位置...\$数	\$1{/N1 は }/N2 を \$1/V3 て \$1/V4.kako . "N1 は"が\$1の位置のいずれかの位置に移動してもよいとする。

### 2.3.1 離散記号

[書式] /「文法属性レベル」

[使用条件] 日本語文型パターンで使用。複数の文法属性ラベルを指定することが可能。

[記述例] /cftk, /cf, /y, /

[説明] 文型パターン以外の要素の挿入可能位置を指定する記号である。文法属性ラベルは以下のような挿入可能な要素の文法的なレベルを表す。

\* 連用節 (/y)

節と節の間に連用節の挿入を認める．同一の節内には挿入を認めない．離散記号 /y で示される．

\* 連体節 (/t)

名詞句の直前に連体節の挿入を認める．同一の名詞句内には挿入を認めない．離散記号 /t で示される．

\* 格要素 (/c)

文型パターン内に存在する格要素の前後に別の格要素の挿入を認める．但し，同一格要素内に別の格要素が挿入されてはならない．また，格要素のないところに格要素を挿入してはならない．離散記号 /c で示される．

\* 連用修飾要素 (/f)

挿入可能な連用修飾要素は，形容詞連用形，副詞，副詞句のいずれかとする．挿入可能な位置は，文頭，述部の直前，形容詞の直前，動詞の直前とする．離散記号 /f で示される．

\* 連体修飾要素 (/k)

挿入可能な連体修飾要素は，連体詞，「A の」・「A と」などの名詞句構成要素，形容詞連体形，動詞連体形のいずれかとし，文型パターン内の名詞句直前への挿入を認める．同一名詞句内への挿入は認めない．離散記号 /k で示される．

[使用例]

(日本語) あの建物はああ見えても新しい。

(英語) That building is still new despite appearance.

(日単P)  $\frac{y}{(1)}$  \$1^{\wedge}\{\frac{cf}{(2)} GEN1 \frac{k}{(3)} N2 \text{は}\} \frac{tcfk}{(4)} \text{ああ} \frac{f}{(5)} (\text{見え|みえ}) \text{ても} \\$1 \frac{yf}{(6)} AJ3。

(英単P) AJ1 N2 be still AJ3 despite appearance.

⇨ (1) 文型の先頭に連用節 (/y) の挿入を認める．

(2) 限定詞の直前に格要素 (/c) と連用修飾要素 (/f) の挿入を認める．

(3) 名詞の直前に連体修飾要素 (/k) の挿入を認める．

(4) 字面「ああ」の直前に連体節 (/t)，格要素 (/c)，連用修飾要素 (/f)，連体修飾要素 (/k) の挿入を認める．

(5) 要素選択記号の直前に (/f) の挿入を認める．

(6) 形容詞の直前に連用節 (/y) と連用修飾要素 (/f) の挿入を認める。

### 2.3.2 文節境界記号

[書式] !

[使用条件] 日本語文型パターンで使用。

[記述例] !

[説明] 文節の境界を表す。入力日本文とのマッチ条件として利用する。

[使用例]

(日本語) 資本主義と社会主義は相入れない概念だ。 (英語) Capitalism is incompatible with socialism. (日単P) /ytcfk N1 と /cfk N2 は /cf 相入れない! <sub>(1)</sub> 概念だ。 (英単P) N1 be incompatible with N2.
---

⇒ (1) !の部分が文節境界を示す。

### 2.3.3 要素選択記号

[書式] #数(パターン記述1 | パターン記述2 | ...)

[使用条件] 日英文型パターン共通。必ずしも日英セットで用いる必要はない。

[記述例] #1(.kako|.genzai), #1(V4.kako|ND4をした), #2(とかく|兎角)

[説明] 複数の選択要素し、照合で複数記述された要素のうちいずれかにマッチすればよいという条件を示す。

[使用例]

(日本語) 彼女はにっこり笑ってぼくにあいさつした。 (英語) She greeted me with a big smile. (日単P) /y \$1^{/tcfk N1 は} /cf にっこり \$1 /f 笑って \$1 /ytck N2 に \$1 /cf (V3.kako ND3をした) <sub>(1)</sub> 。 (英単P) N1 V(V3 ND3) <sub>(1)</sub> ^past N2 ^obj with a big smile.
--

⇒ (1) 「V3.kako」または「ND3をした」のそれぞれいずれかにマッチすればよいことを示す。字面だけではなく、変数や関数も要素選択記号を使用して記述できる。

### 2.3.4 任意要素記号

[書式] # 数 [パターン記述]

[使用条件] 日本語文型パターンで使用，または日英セットで使用する．

[記述例] #4[!ずっと]，#4[/ADV4]

[説明] 照合の際にはあってもなくてもよい要素を表す．日本語にのみある場合と，日英で対応する任意要素がある場合との2つの使用方法がある．各ケースの英語側での訳出方法は以下の通り．

- 日本語のみに任意要素がある場合  
日本語の日本語側の該当表現は英語側では訳出しない．
- 日英で対応する任意要素がある場合  
日本語の任意要素に適合した表現がある場合は，それが英語の任意記号で示された部分に橋渡しされて訳出される．

[使用例]

(日本語) 大阪に着くまでの間、ずっと本を読み続けた。 (英語) I read until I reached Osaka. (日単P) /y </tk N1 は> /tcfk N2 に /cf V3 <sup>rentai</sup> (まで   迄) の ! 間、 <! N4 は> #1[!ずっと] <sub>(1)</sub> /tck 本を /cf V6 (つづけ   続け) た。 (英単P) <I N4> V6 <sup>past</sup> until <I N1> V3 <sup>past</sup> N2.
---

⇒ (1) #1[! ずっと] が任意要素用いた記述要素となり，照合の際にはあってもなくてもよい要素となる．#1[! ずっと] に該当する表現があってもなくても，英語側ではそれに対応する表現を訳出する必要はない．

### 2.3.5 順序任意記号

[書式] # 数 { パターン記述 1 ，パターン記述 2 }

[使用条件] 日本語文型パターンで使用．

[記述例] #1{/N2 が ， /N3 で }

[説明] 照合の際に指定した複数の隣接した要素が入れ替えってもよいとする．

[使用例]

(日本語)	少年は休暇で郷里に帰る途中自動車事故に遭った。
(英語)	The boy was on his way home for the holidays when he met with an auto accident.
(日単P)	/y \$1^{/tcfk N1 は} #1{/tcfk N2 で, /tcfk 郷里に} /cf (帰る   かえる) ! 途中 \$1 /tck N3 に /cf V4.kako 。
(英単P)	$N1 \text{ be}^{\text{past}} \text{ on } N1^{\text{pron}} \text{ poss way home for } N2 \text{ when } N1^{\text{pron}} \text{ V4}^{\text{past}} \text{ with } N3.$

⇒ (1) #1{/tcfk N2 で, /tcfk 郷里に} が順序任意記号を使用した部分となり, 「/tcfk N2 で」と「/tcfk 郷里に」を入れ替えてもよいことを示す。使用例の文型パターンには, 以下のいずれもが適合する。

→ 少年は 郷里に休暇で 帰る途中自動車事故に遭った。

→ 少年は 休暇で郷里に 帰る途中自動車事故に遭った。

### 2.3.6 位置変更可能記号

[書式] \$数^{ {パターン記述} 移動先位置...\$数

[使用条件] 日本語文型パターンで使用。

[記述例]  $\$1^{\{/N1 \text{ は} \}}/N2 \text{ を} \$1/V3 \text{ て} \$1/V4.kako.$

[説明] 照合の際に要素の移動可能な位置を指定する。

[使用例]

(日本語)	兄はわたしを脅して両親にうそをつかせた。
(英語)	My brother intimidated me into telling a lie to our parents.
(日単P)	/y \$1^{/tcfk N1 は} /tcfk N2 を /cf V3 (て   で) \$1_{(1)} #1{/ytk N4 に, /tcfk N5 を} /cf (V6.sase.kako   V6^sase.kako) 。
(英単P)	$N1 \text{ V3}^{\text{past}} \text{ N2}^{\text{obj}} \text{ into } V6^{\text{ing}} \text{ N5 to } N4.$

⇒ (1) 「/tcfk N1 は」が\$1の位置のいずれかの位置に移動してもよいということを示す。使用例の文型パターンには, 以下のいずれもが適合する。

→ 兄は わたしを脅して両親にうそをつかせた。

→ わたしを脅して 兄は 両親にうそをつかせた。



## 2.4 関数について

本稿で取り扱う関数について説明する。関数は、表記上の揺らぎを吸収し、適用範囲の広い文型パターンを記述するために用いられている。文型パターンに使用されている関数の中から、日本語文型パターンに用いられており、汎用性の高いことが期待される2つの関数(42種類)を取り扱う(表5)[5]。以下に、取り扱うそれぞれの関数について説明する。

### 2.4.1 日本語様相関数

[書式] .関数名

[使用条件] いずれも日本語文型パターンで使用。

[記述例] *V2.kako* , *N3.da*

[説明] 変数に付属した助動詞相当表現に該当し、主に「時制(tense)」、「相(aspect)」、「態(voice)」、「様相(modality)」の4種類がある。日本語様相関数には、*.kaishi*(*V*出す,*V*始める),*.gimu*(*V*なければならない,*V*なければならぬ,...)など、同じ意味をもつ複数の表現の表記のバリエーションを吸収するために使用されているものや、(*.genzai.kako*)など同じタイプの機能を持つ様相表現をOR条件で記述する際に使用するものや、*.masu* , *.noda*など直接英語と対応する表現がなく、一律任意化を行うことを目的として使用しているものがある。

#### \* 時制関数

主にある時点からみて現在・過去・未来のどれにあたるかといった時間的關係を表す関数である。*.genzai* , *.kako* はそれぞれ、英語側の語形関数である<sup>^</sup>*present* , <sup>^</sup>*past* に対応する。

#### \* 相関数

出来事が瞬間的なのか、継続的か、断続的か、反覆するのか、やがて終わるのかといった全過程のどの局面にあるのかを表す関数である。英語側は対応する関数を定義しておらず、日本語文型パターンでのみで使用する。

#### \* 態関数

表現されている出来事をどの視点に立って見るかを表す関数である。授受動作のどの視点に立ってみるかを表す「てくれる」などの受益表現や「てやる」などの与益表現も含む。

表 5: 使用する関数の機能と記述形式

統語的・意味的分類		日本語		英語				
統語的	意味的	表記	適合表現	表記	適合表現			
時制 tense	現在	.genzai	V	.present	動詞基本形			
	過去	.kako	V た, V だ	.past	動詞過去形			
相 aspect	進行・結果・反復・状態など (動詞のタイプによる)	.teiru	V ている, V ている	なし	-			
	完了相(否定的)	.teshimau	V てしまう					
	てある	.tearu	V である, V である					
	ておく	.teoku	V ておく, V ておく					
	てくる	.tekuru	V てくる, V てくる					
	ていく	.teiku	V ていく, V ていく					
	開始相	.kaishi	V はじめる, V 出す					
	試行動作	.temiru	V てみる, V てみる					
	状態変化	.joutaihenka	AJ くなる, AJV になる					
	状態変化+使役	.nisuru	AJ くる, AJV にする					
態 voice	使役態	.sase	V せる, V させる	.passive, .can	be V.ed, can			
	受動態・可能態	.rareru	V れる, V られる					
	可能態	.dekiru	V できる					
	やり ・もらい	.tekureru .teyaru	V てくれる, V てくれる V てやる, V てやる			なし	-	
否定		.hitei(nai)	ない, ぬ, くない	.not	not			
様相 modality	命題 に対する 認識	真偽 判断	証拠性判断	.suitei	ようだ, らしい, みたい	なし	-	
			伝聞	.souda	V (連用形) そうだ(推量) V (連体形) そうだ(伝聞)			
		価値 判断	断定	.da .desu	だ, である です			
			義務	.gimu	V べき, V なくてはならない, V なくてはならぬ, V なくてはいけない, V ないといけない, V ねばならない, V なければならない, V ねばならぬ, V ねばいけない, V なければならない, V なければいけない			.must, .should
		選択的 判断	.hougayoi	方がよい, 方がいい	.better			had better
	話し手 の態度	許可	.teyoi	V てよい	.may	may		
		希望	.tai	V たい	なし	-		
		意志	.you	V よう	.will	will		
		意志+試行動作	.utosuru	V うとすると, V ようとすると	.will, .would	will, would		
		予定	.yotei	V ことにする, V つもり	なし	-		
		聞き手 への働きかけ	依頼	.tekudasai	V てください	.please	please	
			命令	.meireigo	V なさい, V たまえ	なし	-	
			禁止	.kinshi	V するな	.donot	don't, do not	
	その他	断定・説明	.noda	のだ, のである	なし	-		
	丁寧	.masu	ます					
その他	過剰	.sugiru	すぎる	.too	too			
不明	推量・判断	.darou	だろう	.will, .may	Will, may			
	受動態・可能態	.reru	V れる, V られる	.passive, .can	be V.ed, can			
	断定	.da	だ, である	なし	-			
	推定	.rashii	らしい					
	丁寧	.teinei	ます					
	破綻	.kirezuru	V きれぬ, V きれず, V きれない					

< 同値型グループ関数 >

名称	表記	適合表現
体言述部型(です・だ型)	.#da	N だ, N である, N です
推量型(だろう・でしょう型)	.#darou	V だろう, V でしょう

- \* 否定関数  
否定の意味を表す関数である。
- \* 様相関数  
相手(文)が指す内容に対する話し手の判断や心的態度を表す関数である。
- \* その他の関数  
過剰な程度を表す補助動詞の「過ぎる」と英語の「too」を表す関数である。

[使用例]

(日本語)	私の胸からかげりが一掃されたように感じた。
(英語)	I felt as if a cloud had been lifted from my heart.
(日単P)	/y #1{/tcfk N1 の /k N2 から,/tcfk N3 が } /cf V4 <u>.reru</u> <sub>(1)</sub> <u>.kako</u> <sub>(2)</sub> <u>.suitei</u> <sub>(3)</sub> #3( <u>.kako</u> <sub>(4)</sub>   <u>.genzai</u> <sub>(4)</sub> ) /yf V5 #2( <u>.kako</u> <sub>(5)</sub>   <u>.genzai</u> <sub>(5)</sub> )。
(英単P)	N1 V5#2(^past ^present) as if N3 had been V4#3(^past ^present) from N1^poss N2.

- ⇒ (1) 動詞の直後に態関数 *.reru* 「れる, られる」が必要となる。
- (2) 態関数 *.reru* の直後に時制関数 *.kako* 「た, だ」が必要となる。
- (3) 時制関数 *.kako* の直後に様相関数 *.suitei* 「ようだ, らしい」が必要となる。
- (4) 様相関数 *.suitei* の直後に時制関数 *.kako* 「た, だ」, または, 時制関数 *.genzai* が必要となる。
- (5) 動詞の直後に時制関数 *.kako* 「た, だ」, または, 時制関数 *.genzai* が必要となる。

## 2.4.2 同値型グループ関数

[書式] *.#* グループ関数名

[使用条件] 日本語文型パターンで使用。

[記述例] *V2.#da* , *N3.#darou*

[説明] 日本語様相関数の記述を拡張したもので, 英語文型に影響を与えない範囲で, 同種の意味の日本語様相関数をまとめて指定する機能を持つ。これにより, 日英文型パターン作成の元になった標本文で使用されていた日本語様相関数だけでなく, 同値型グループ関数として定義された他の関数にも適合させることが可能となる。

\* 体言述部関数

「.da」「.desu」を1つにまとめた関数を表す。V3(.da|.desu)と意味的に同等となる。

\* 推量型関数

「.darou」「.desho」を1つにまとめた関数を表す。V3(.darou|.desho)と意味的に同等となる。

[使用例]

(日本語) 門の側に大きな木のある家です。

(英語) It's a house with a large tree right beside the gate.

(日単P) /ytcfk N1 の /k 側に /cf AJ2^rentai /f N3 の /f (ある | 在る | 有る) ! N4.#da<sub>(1)</sub>。

(英単P) It is N4 with AJ2 N3 right beside N1.

⇒ (1) 「です」だけでなく、「だ」や「である」にもマッチするようになる。

### 3 評価パラメータ

#### 3.1 網羅性の評価

文型パターンは、必ずしも入力文のすべての要素が適合する文型パターン（以後、適合パターンと呼ぶ）である必要はなく、入力文の主要な構造が適合し意味的に正しい文型パターンであればよい。検索ツールは、入力文に対して、当該文型パターンのすべての要素が、定義された順に出現する文型パターンを、適合パターンとして抽出するので、適合パターンは、以下の2種類に分類できる。

<完全一致文型>：入力文のすべての要素が文型パターンの要素と適合する文型パターン

<部分一致文型>：入力文の一部の要素が文型パターンに定義されない要素となる文型パターン

従って、適合パターンとはいえ、必ずしも入力文のすべての要素が解釈を与えるものではないので、適合パターンについて、それが入力文の何%をカバーしているか（網羅性）が問題となる。

そこで本稿では、網羅性を評価するパラメータとして、[3]で示されている評価パラメータを使用する。以下に、概略を説明する。

### 3.1.1 文型再現率 $R1$

$R1$  は「全入力文のうち、適合する文型パターンの存在した入力文の割合」を表し、以下の式で定義される。

$$R1 = M/I \quad (1)$$

$I$  : 全入力文の数

$M$  : 適合パターンがある入力文の数

### 3.1.2 適合文型数 $N$

$N$  は「入力文に対する適合パターン数の平均値」を表し、以下の式で定義される。

$$N = \sum_{i=1}^M N_i/M \quad (2)$$

$N_i$  :  $i$  番目の入力文の適合パターンの数

### 3.1.3 単語一致率 $R2\_W$

$R2\_W$  は「入力文の単語単位に見た再現率」を表すもので、以下の式で定義される。

$$R2\_W\_all = \sum_{j=1}^M S_j / \sum_{i=1}^I L_i \quad (3)$$

$$R2\_W\_match = \sum_{j=1}^M S_j / \sum_{i=1}^M L_i \quad (4)$$

$S_j$  : 「最尤文型パターン」が解釈できる単語数

$L_i$  :  $i$  番目の入力文の単語数

「最尤文型パターン」とは、適合パターンのうち、最も広範囲に入力文と一致する文型パターンを言う。

### 3.1.4 $R1$ , $N$ , $R2\_W$ の求め方

図1を用いて、文型再現率  $R1$ 、適合文型数  $N$ 、および、単語一致率  $R2\_W$  の求め方を示す。まず入力文1~3と文型パターンを照合し、適合する文型パターンを抽出する。入

入力文1には適合パターンが2パターン存在して、入力文2は適合パターンが存在せず、入力文3には適合パターンが1つ存在した。入力文3文中、2文は適合パターンが存在するので、この場合、 $R1$ は、「適合パターンがある入力文数」割る「全入力文数」で求まり、 $R1 = 2 / 3 \times 100 = 66.70(\%)$ となる。

また、適合パターンが合計3つ存在するので、 $N$ は「適合パターン数」割る「適合パターンがある入力文数」で求まり、 $N = 2 / 3 = 0.66$  (パターン)となる。

次に、入力文1の単語数は10単語で、入力文1に適合する文型パターンで、最も広範囲に入力文と一致する文型パターンはパターン番号1で、解釈できる単語数は10単語となる。また入力文2の単語数は11単語で、適合パターンは存在しない。入力文3の単語数は9単語で、入力文3に適合する文型パターンで、最も広範囲に入力文と一致する文型パターンはパターン番号3で、解釈できる単語数は9単語となる。以上より、 $R2\_W\_all$ は、「最尤文型パターンの単語数」割る「全入力文の単語数」で求まり、 $R2\_W\_all = (10 + 9) / (10 + 9 + 11) \times 100 = 63.33(\%)$ となる。また $R2\_W\_match$ は、「最尤文型パターンの単語数」割る「適合パターンがある入力文の単語数」で求まり、 $R2\_W\_match = (10 + 9) / (10 + 9) \times 100 = 100.00(\%)$ となる。ただし、実験上では、文型パターンは入力文から作成されるため、当該入力文(以後、自己文と呼ぶ)から作成された文型パターン(以後、自己パターンと呼ぶ)は、自己文に対して使用しないこととする。

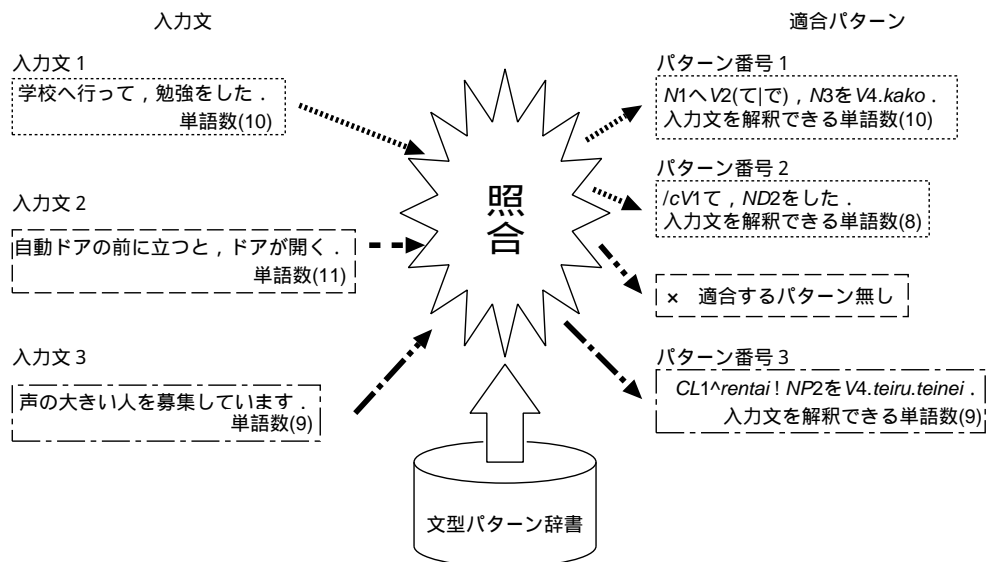


図 1:  $R1$  ,  $N$  ,  $R2\_W$  の算出方法

### 3.2 パターンの増加数への換算

文型再現率  $R1$  や適合文型数  $N$  は、文型パターン辞書の規模（文型パターン数）に対して単調増加する。文型パターン数  $p$  の文型パターン辞書における  $R1$  の曲線は次式で表される [4]。

$$R1 = (1 - \exp(-\lambda_1(p)^{\lambda_2})) \times 100.00(\%) \quad (5)$$

[1] の文型パターン辞書の記述子に変更を加えると、入力文と文型パターンとの適合の仕方が変化して、文型再現率  $R1$  や適合文型数  $N$  が変化する。式 (5) から  $p$  を表す逆関数を求め、その変化後の  $R1$  を代入すると、記述子に変更を加えられた文型パターン辞書の規模の換算値を求めることができる。

### 3.3 文型パターン拡大率 $\eta$

文型パターン拡大率  $\eta$  は「評価対象の文型パターン辞書が、基準となるパターン辞書の文型パターン数に換算して、何倍に相当するか」を表し、以下の式で定義される。

$$\eta = X/B \quad (6)$$

$B$  : 基準となる文型パターン辞書の文型パターン数

$X$  : 対象となる文型パターン辞書の文型パターン数の換算値

本稿では、「文型再現率  $R1$ 」からみた「文型パターン拡大率  $\eta_{R1}$ 」を使用する。このとき、換算値  $X$  は式 (5) から求める。

$\eta_{R1}$  の推定方法

基準となるパターン辞書を用いた実験から、文型再現率  $R1$  と文型パターン数  $p$  の関係を図で示す（図 2：単語レベル，図 3：句レベル，図 4：節レベル，図 5：全レベル）。図中の縦軸は、文型再現率  $R1$  を示し、横軸は文型パターン数  $p$  を示す。サンプル点（×）は文型パターン数と  $R1$  の実測値を示す。非線形回帰分析より、文型再現率  $R1$  の文型パターン数  $p$  に対する特性を、式 (5) で近似する。近似曲線を点線で示す。

これより、基準となるパターン辞書の実測値より、 $\lambda_1$ 、 $\lambda_2$  を求め、対象となるパターン辞書の実測値  $R1$  より、 $p_{R1}$  を求める。そして、式 (6) にそれぞれ代入することで、 $\eta_{R1}$  を求める。それぞれのレベルごとの係数は以下の通りである。

- 単語レベル  
 $\lambda_1 = 0.00407944$  ,  $\lambda_2 = 0.47791782$
- 句レベル  
 $\lambda_1 = 0.03371123$  ,  $\lambda_2 = 0.36199351$
- 節レベル  
 $\lambda_1 = 0.37352190$  ,  $\lambda_2 = 0.11193710$
- 全レベル  
 $\lambda_1 = 0.53517855$  ,  $\lambda_2 = 0.01437754$



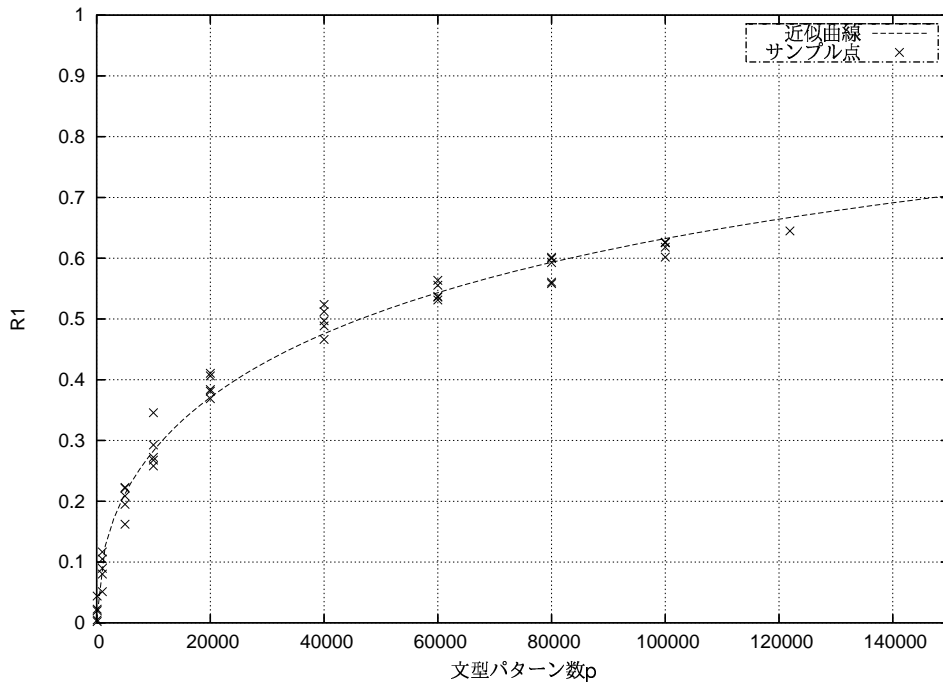


図 2:  $R1$  と文型パターン数の関係図 (文法・単語レベル)

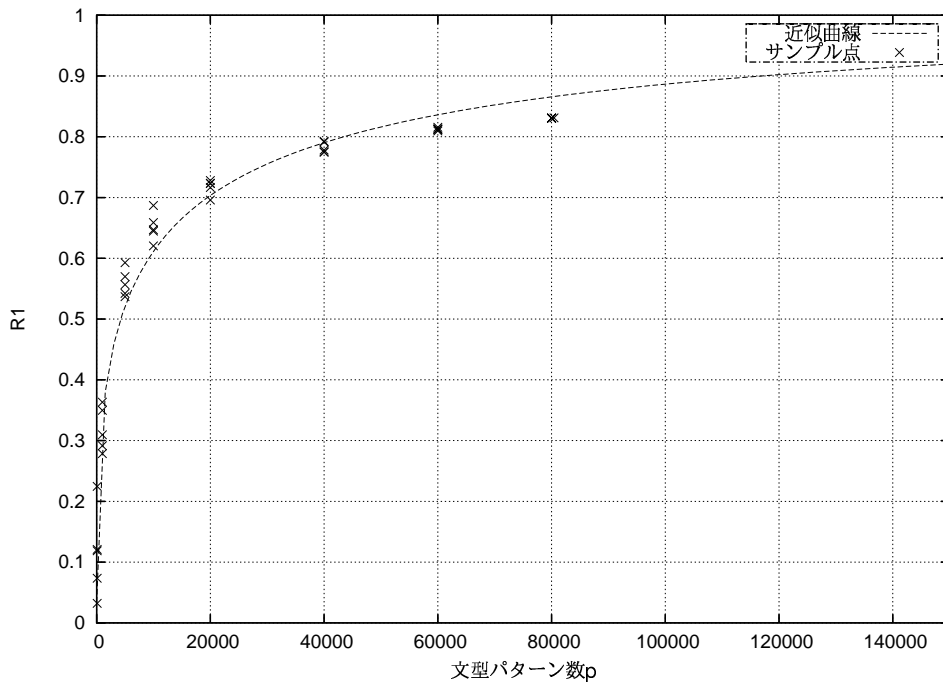


図 3:  $R1$  と文型パターン数の関係図 (文法・句レベル)

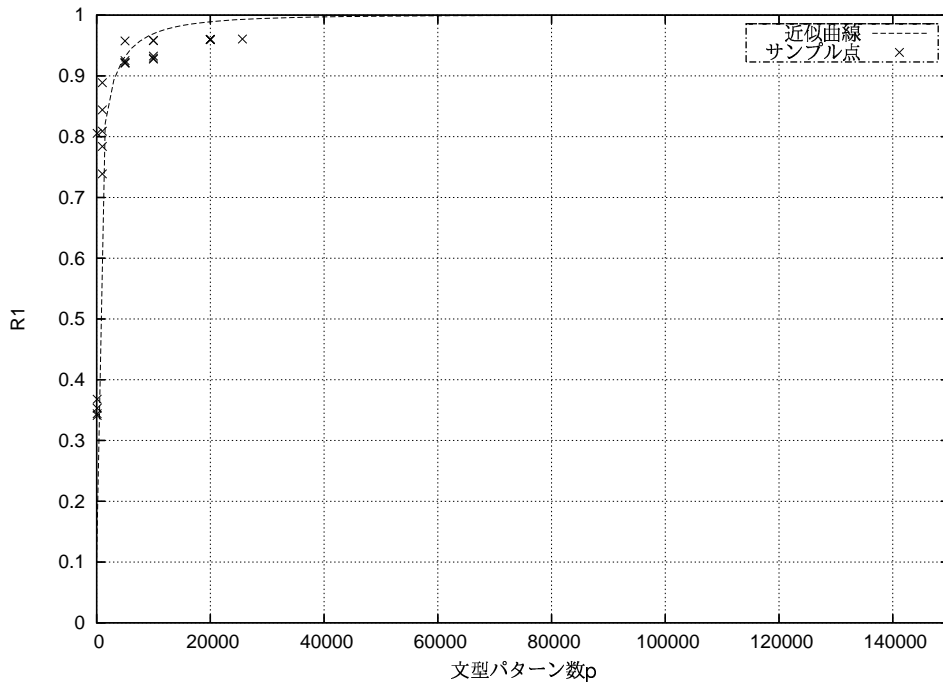


図 4:  $R1$  と文型パターン数の関係図 (文法・節レベル)

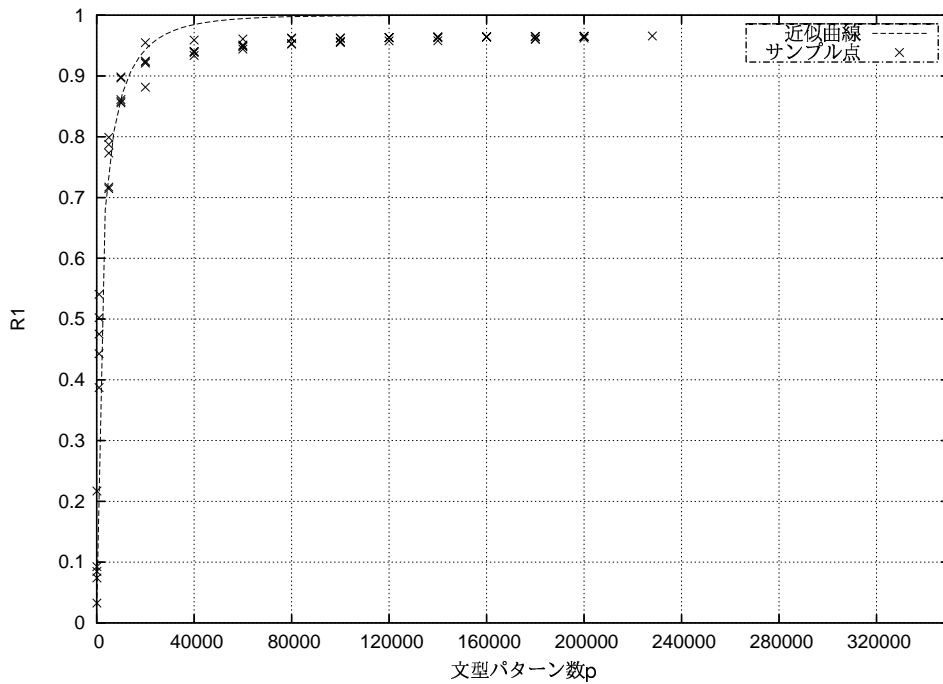


図 5:  $R1$  と文型パターン数の関係図 (文法・全レベル)

## 4 汎化効果の評価

### 4.1 汎化効果の計測方法

#### 4.1.1 基準パターン

[1]の文型パターン辞書(217,703パターン)から,単語・句・節レベルの文型パターン集を抽出し,それを「基準パターン辞書」として,記号・関数の汎化効果を評価する.基準パターン辞書は,変数の汎化レベルごとに番号(①~④)を付ける.

- ① : 単語レベル(119,229パターン, (①/④)×100.00=54.77%)
- ② : 句レベル (75,708パターン, (②/④)×100.00=34.78%)
- ③ : 節レベル (22,766パターン, (③/④)×100.00=10.46%)
- ④ : 全レベル (217,703パターン, (④/④)×100.00=100.00%)

#### 4.1.2 記号・関数の削除

基準パターン辞書には,既に記号・関数が導入されている.記号・関数の各種類ごとに文型パターンの汎化効果を評価するためには,基準パターン辞書の汎用性の評価結果と記号・関数を削除した文型パターン辞書の汎用性の評価結果を比較する必要がある.そのために,基準パターン辞書から,記号・関数を削除した文型パターン辞書を作成する(それぞれ作成した文型パターン辞書ごとに番号①から⑥を割り振る).

記号の削除方法は,文節境界記号以外の記号は,自己文に適合するように基準パターン辞書から記号の削除を行う.文節境界記号は基準パターン辞書から記号の削除を行う.関数の削除方法は,関数は基準パターン辞書から関数の削除を行う.以下に,基準パターン辞書から記号・関数を削除する様子を示す.

離散記号の削除 2.3.1節で説明した離散記号を基準パターン辞書から削除して,新たな文型パターン辞書(番号①)を作成する.

[削除前]	$\boxed{/y}$ $\langle \boxed{/tk}$ N1 は $\rangle \boxed{/tcfk}$ N2 の $\boxed{/k}$ N3 が $\boxed{/tcfk}$ ああ $\boxed{/f}$ AJ4 とは $\boxed{/yf}$ V5.hitei.kako。
	↓ 自己文に当たるように削除
[削除後]	$\boxed{\phantom{/y}}$ $\langle \boxed{\phantom{/tk}}$ N1 は $\rangle \boxed{\phantom{/tcfk}}$ N2 の $\boxed{\phantom{/k}}$ N3 が $\boxed{\phantom{/tcfk}}$ ああ $\boxed{\phantom{/f}}$ AJ4 とは $\boxed{\phantom{/yf}}$ V5.hitei.kako。

文節境界記号の削除例 2.3.2 節で説明した文節境界記号を基準パターン辞書から削除して、新たな文型パターン辞書（番号②）を作成する。

[削除前]	$/y$ $\$1^{\boxed{/tcfk}}$ N1 は } #1 $\boxed{/tcfk}$ TIME3 $\boxed{!}$ 相づちを $\boxed{!}$ 打ちながら $\$1$ $/ytck$ N4 の $/k$ N5 を $/cf$ V6.kako。
	↓
[削除後]	$/y$ $\$1^{\boxed{/tcfk}}$ N1 は } #1 $\boxed{/tcfk}$ TIME3 $\boxed{\phantom{!}}$ 相づちを $\boxed{\phantom{!}}$ 打ちながら $\$1$ $/ytck$ N4 の $/k$ N5 を $/cf$ V6.kako。

要素選択記号の削除例 2.3.3 節で説明した要素選択記号を基準パターン辞書から削除して、新たな文型パターン辞書（番号③～⑥）を作成する..ただし本稿では、要素選択記号の中で「時制を汎化しているとき（対応型選択要素も含む）」、「ND（用言性名詞）」が使われているとき、「その他」の3通りに場合分けする。

- (.kako|.genzai) の場合

[削除前]	$/y$ $\langle \boxed{/tk}$ N1 は $\rangle \boxed{/tcfk}$ 目を $\boxed{/cf}$ 覚ましたら $\boxed{/ytck}$ (夜 よる よ) が $\boxed{/cf}$ 明ける' と ころだ' $\boxed{\#1 (.kako .genzai)}$ 。
	↓ 自己文に当たるように削除
[削除後]	$/y$ $\langle \boxed{/tk}$ N1 は $\rangle \boxed{/tcfk}$ 目を $\boxed{/cf}$ 覚ましたら $\boxed{/ytck}$ (夜 よる よ) が $\boxed{/cf}$ 明ける' と ころだ' $\boxed{.kako}$ 。

- (V...|ND...) の場合

[削除前] /y \$1^{/tcfk N1 は } /cf (V2|ND2 をし) (て|で) \$1 /ytck ND3 を /cf している。  
 ↓ 自己文に当たるように削除  
 [削除後] /y \$1^{/tcfk N1 は } /cf V2 (て|で) \$1 /ytck ND3 を /cf している。

• (その他) の場合

[削除前] /y \$1^{/tcfk N1 は } /tcfk (どこ|何処) に /cf V2.kako か /yf V3.hitei が \$1 /yf (V4.kako|ND4 をした)。  
 ↓ 自己文に当たるように削除  
 [削除後] /y \$1^{/tcfk N1 は } /tcfk 何処に /cf V2.kako か /yf V3.hitei が \$1 /yf (V4.kako|ND4 をした)。

任意要素記号の削除例 2.3.4 節で説明した任意要素記号を基準パターン辞書から削除して、新たな文型パターン辞書(番号⑦)を作成する。

[削除前] /y \$1^{/tcfk N1 は } #1 [/tcfk TIME3] !うなずきながら \$1 /ytck N4 の /k N5 を /cf V6.kako 。  
 ↓ 自己文に当たるように削除  
 [削除後] /y \$1^{/tcfk N1 は } /tcfk TIME3 !うなずきながら \$1 /ytck N4 の /k N5 を /cf V6.kako 。

順序任意記号の削除例 2.3.5 節で説明した順序任意記号を基準パターン辞書から削除して、新たな文型パターン辞書(番号⑧)を作成する。

[削除前] /y \$1^{/tcfk N1 は } #1 {/cf #2 [GEN3] /k N4 に \$1,/tcfk N5 が } /cf (き|来) て \$1 /ycf V6.kako 。  
 ↓ 自己文に当たるように削除  
 [削除後] /y \$1^{/tcfk N1 は } /cf #2 [GEN3] /k N4 に \$1,/tcfk N5 が /cf (き|来) て \$1 /ycf V6.kako 。

位置変更可能記号の削除例 2.3.6 節で説明した位置変更可能記号を基準パターン辞書から削除して，新たな文型パターン辞書（番号⑨）を作成する．

[削除前]	/y	$\boxed{\$1^{\wedge}\{/cf\ GEN1 /k\ N2\ \text{は}\}}$	/tcfk ああ /f(見え みえ) ても	$\boxed{\$1}$	/yf AJ3。
$\Downarrow$ 自己文に当たるように削除					
[削除後]	/y	$\boxed{\{/cf\ GEN1 /k\ N2\ \text{は}\}}$	/tcfk ああ /f(見え みえ) ても	$\boxed{\quad}$	/yf AJ3。

日本語様相関数の削除例 2.4.1 節で説明した日本語様相関数を基準パターン辞書から削除して，新たな文型パターン辞書（番号⑩～⑪）を作成する．

[削除前]	/y	</tk N1 が>	/tcfk N2 を	/cf V3	$\boxed{.kako}$	$\wedge$ katei	/ytck N4 が	/cf	AJ5	$\boxed{.joutaihenka}$	#1	(	$\boxed{.kako}$	$\boxed{.genzai}$	)。	
$\Downarrow$																
[削除後]	/y	</tk N1 が>	/tcfk N2 を	/cf V3	$\boxed{\quad}$	$\wedge$ katei	/ytck N4 が	/cf	AJ5	$\boxed{\quad}$	$\boxed{\quad}$	$\boxed{\quad}$	$\boxed{\quad}$	$\boxed{\quad}$	$\boxed{\quad}$	。

同値型グループ関数の削除例 2.4.2 節で説明した同値型グループ関数を基準パターン辞書から削除して，新たな文型パターン辞書（番号⑫～⑭）を作成する．

[削除前]	/ytcfk N1 は	/cf AJ2 が	/ytfk N3 N4	$\boxed{.#da}$	。
$\Downarrow$					
[削除後]	/ytcfk N1 は	/cf AJ2 が	/ytfk N3 N4	$\boxed{\quad}$	。

#### 4.1.3 計測方法

計測には，実際に入力文と文型パターンを照合して，第3章で示した評価パラメータを求める．以下に計測条件を示す．

(1) 変数の意味的な制約条件は無視する．

入力文に適合する文型パターンの中から意味的に適切な文型パターンを選択する方法は別の課題と考え，ここでは，得られた文型パターンの潜在的な能力を評価する．

(2) クロスバリデーション型<sup>1</sup>の実験 .

計測用の入力文は , 対訳標本からランダムに選んだ 10,000 文 (=  $I$ ) とする . 文型パターンを作成するために用いた文であるが , 入力文から作られた文型パターンは , 照合に用いないので , クロスバリデーション型の実験である .

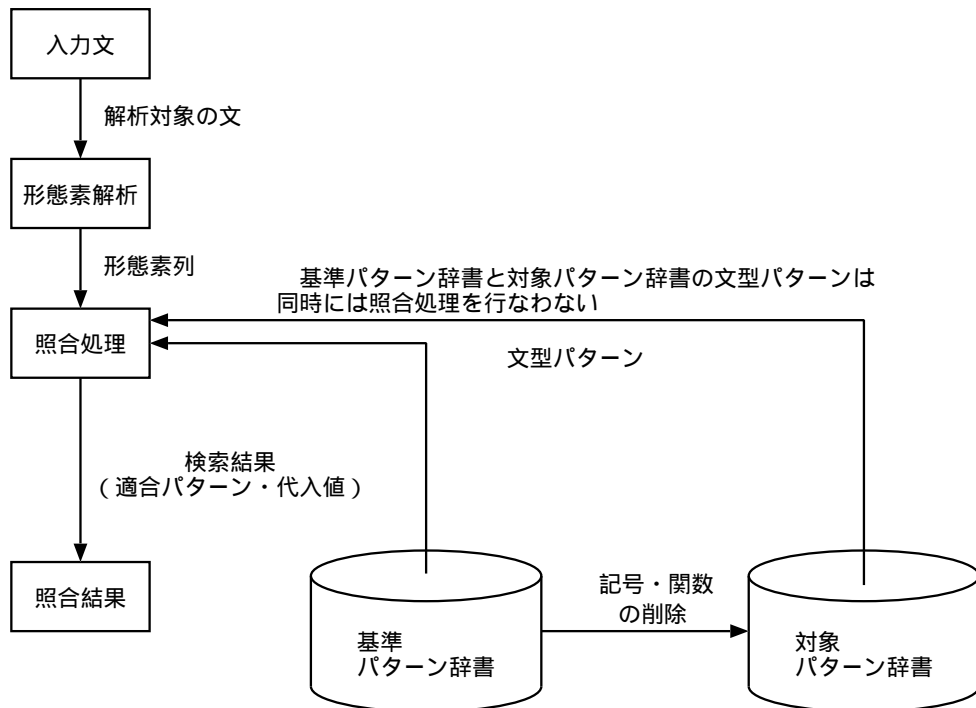
クロスバリデーション型の実験を行なうための諸条件を以下に示す .

#### 諸条件

検索ツール : spm\_prot39[6]  
入力形態素 : all\_keitaiso+zokusei\_20050613toku5 ( 123,618 文 ) からランダムに 10,000 文  
文型パターン辞書 : 基準パターン辞書 ,  
上記で作成したそれぞれの記号・関数を削除した文型パターン辞書 ( 対象パターン辞書 )  
線形要素判定規則 : rule\_20060710

<sup>1</sup>標本を使用した学習において , 標本の一部を残して学習し , 残した標本をテストに使用することを繰り返し , 評価の信頼性を上げる方法

#### 4.1.4 計測の様子



検索ツール [6] を使用した時の具体的な照合実験の様子を以下に示す。

(1) 照合実験に必要なデータベース，判定規則および，プログラムを用意する。

→ データベース：入力文の形態素解析結果 (all\_keitaiso+zokusei\_20050613toku5)

入力文：彼のお母さんがああ若いとは思わなかった。

▼ 形態素解析結果



INPUT=AC000004-00=

彼のお母さんがああ若いとは思わなかった。

1. /彼 1710,({ NI:23,NI:48,KR:9900k08,KR:9901s81,KR:0601s10,KR:9901s86,IM:11131,IM:11211})
2. +の (7410)
3. /お母さん (1100,{NI:80,NI:49,IM:11212,IM:11220})
4. +が (7410)
5. /ああ (1110)
6. /若い (3106,{NY:5,KR:8803c00,IY:A300})
7. +と (7420)
8. +は (7530)
9. /思わ (2392, 思う, 思わ,{NY:32,NY:31,KR:0601a01,KR:1500a00,IY:1200, IY:1262,IY:2111})
- 10.+なかつ(7184, ない, なかつ)
- 11.+た (7216)
- 12.+。 (0110)
- 13./nil

→ 線形要素判定規則 ( rule\_20060710 )

→ 文型照合プログラム ( spm\_prot39 )

(2) 線形要素判定規則のオートマトン ( マップファイル ) を作成する .

% rule2linear.rb ⊂ rule\_linear ⊂ linear.ad ⊂ linear.sd ( 判定規則の A 型変数 , S 型変数 )

% atype2map ⊂ linear.ad ⊂ linear.am ( 判定規則の A 型変数のマップファイル )

% stype2map ⊂ linear.sd ⊂ linear.sm ( 判定規則の S 型変数のマップファイル )

(3) 基準パターン辞書 , および , 対象パターン辞書のオートマトン ( マップファイル ) を作成する .

% pat2atype ⊂ bwjall.pt ⊂ bwjall.pd ( 文型パターン辞書のオートマトン )

% atype2map ⊂ bwjall.pd ⊂ bwjall.pm ( 文型パターン辞書のマップファイル )

(4) 照合

% walker21 ⊂ linear.sm ⊂ linear.am ⊂ bwjall.pm ⊂ -cross ⊂ Mrp/AC/000/004-00.mrp  
( 形態素ファイル名 )

Mrp/AC/000/004-00.mrp.rslt (結果ファイル)

(5) 照合結果の出力

照合結果の出力ファイルの形式の使用を以下に示す。

1. INPUT = 「入力形態素ファイル名」
2. 入力文
3. 形態素情報
4. 入力情報の区切り
5. 記号 (-----)
6. PATTERN = 「適合パターン番号」 = 「適合した要素」 = 「適合した要素の番号」 = 「適合した単語数」
7. 変数名 = 「適合位置」 = 「適合位置」 = 「適合した単語数」

▼ 照合結果の出力ファイルを以下に示す。

INPUT=AC000004-00=

彼のお母さんがああ若いとは思わなかった。

1. /彼 1710,({ NI:23,NI:48,KR:9900k08,KR:9901s81,KR:0601s10,KR:9901s86,IM:11131,IM:11211})
2. +の (7410)
3. /お母さん (1100,{NI:80,NI:49,IM:11212,IM:11220})
4. +が (7410)
5. /ああ (1110)
6. /若い (3106,{NY:5,KR:8803c00,IY:A300})
7. +と (7420)
8. +は (7530)
9. /思わ (2392, 思う, 思わ,{NY:32,NY:31,KR:0601a01,KR:1500a00,IY:1200, IY:1262,IY:2111})
- 10.+なかつ(7184, ない, なかつ)
- 11.+た (7216)
- 12.+。(0110)
- 13./nil

-----  
PATTERN=WJBB081766-00=[N2,AJ3, とは,V4,.hitei,.kako,。 ]=[5,6,7,8,9,10,11,12]=8

N2=[5]=5=1

AJ3=[6]=6=1

V4=[9]=9=1

-----  
PATTERN=WJBB112090-01=[ADV3,AJ4, とは,V5,.hitei,.kako,。 ]=[5,6,7,8,9,10,11,12]=8

ADV3=[5]==1

AJ4=[6]=6=1

V5=[9]=9=1

#1=[ADV3]=[5]

-----  
PATTERN=WJBB112090-02=[ADV2,AJ3, とは,V4,.hitei,.kako,。 ]=[5,6,7,8,9,10,11,12]=8

ADV2=[5]==1

AJ3=[6]=6=1

V4=[9]=9=1

=====

(6) 文型再現率  $R1$  の算出

照合結果ファイルより、適合パターンが抽出された入力文の数を算出する。そして、3.1.1 節の方法によって  $R1$  を求める。なお、入力文から作成されたパターン「自己パターン」は、適合パターンには含めないこととする。

(7) 適合文型数  $N$  の算出

照合結果ファイルより、抽出された適合パターンの数を算出する。ここで、適合パターン数の計算において、注意点が2点ある。1つ目は、 $R1$  の算出の際と同様に、自己パターンは含めないということである。2つ目は適合パターン数の数えかたで

ある．適合パターン数には変数のバインド情報が含まれている．同一の適合パターンで複数の適合の仕方がある場合，各変数や関数の対応する形態素が適合パターンごとに異なることで，適合パターンの個数を数える．これより，3.1.2 節の方法によって  $N$  を求める．

(8) 文型パターン拡大率  $\eta$  の算出

求めた  $R1$  の値と 3.3 節の方法で，文型パターン辞書の文型パターン拡大率  $\eta_{R1}$  を計測する．汎化機能のある記述子を削除するため， $\eta$  は 1 より小さい場合に汎化の寄与があるといえる．

(9) 文種別ごとの調査

照合実験で適合した入力文から文種別ごとに  $R1$  を集計する．

- (i) 文種別 0 : 未分類，または，下記以外の文
- (ii) 文種別 1 : 文接続 1 箇所を持つ文
- (iii) 文種別 2 : 文接続 2 箇所を持つ文
- (iv) 文種別 3 : 埋め込み文 1 つを含む文
- (v) 文種別 4 : 埋め込み文 2 つを含む文
- (vi) 文種別 5 : 文接続 1 箇所 + 埋め込み文 1 つの文

(10) 単語一致率  $R2\_W\_all$  の算出

入力文の形態素解析結果より，入力文の単語数を算出する．また照合結果ファイルより，抽出された適合パターンの数を算出する．ここで，適合パターンの単語数の計算において，注意点が 2 点ある．1 つ目は， $R1$  の算出の際と同様に，自己パターンは含めないということである．2 つ目は適合文型パターンの単語数の数えかたである．複数の適合パターンがある場合，それぞれの適合パターンの単語数の最大値を，その入力文における適合パターンの単語数とする．これより，3.1.3 節の方法によって  $R2\_W\_all$  を求める．

(11) 単語一致率  $R2\_W\_match$  の算出

照合結果ファイルより，適合パターンが抽出された入力文の単語数を算出する．また抽出された適合パターンの数を算出する．ここでも (10) と同様に適合文型パターンの数を算出する．これより，3.1.3 節の方法によって  $R2\_W\_match$  を求める．

## 4.2 汎用性の評価結果

記号・関数における評価パラメータの計測結果と  $\eta_{R1}$  を算出するときに求めた換算値  $X$  (3.3節) を, 変数の汎化レベルごとに分け, 表6~9にまとめる(各種類ごとの単語・句・節・全レベルの計測結果は付録3の表1~24にまとめる)。

表6: 評価パラメータの結果(文法・単語レベル)

評価項目		$R1$	$N$	$R2\_W\_all$	$R2\_W\_match$	$\eta_{R1}$	換算値 $X$
文型パターン辞書							
❶	基準パターン	60.18	36.71	42.23	74.49	1.00	119,229
①	全ての離散記号	<u>22.19</u>	<u>14.37</u>	<u>17.29</u>	<u>100.00</u>	0.07	8,346
②	文節境界記号	<u>60.23</u>	36.71	<u>42.26</u>	<u>74.47</u>	1.00	119,229
⑥	全ての要素選択記号	<u>59.21</u>	<u>32.07</u>	<u>41.33</u>	<u>74.13</u>	0.95	113,268
⑦	任意要素記号	<u>54.38</u>	<u>35.07</u>	<u>38.51</u>	<u>75.85</u>	0.72	85,845
⑧	順序任意記号	<u>60.10</u>	<u>36.53</u>	<u>42.10</u>	<u>74.40</u>	0.99	118,037
⑨	位置変更可能記号	<u>59.90</u>	<u>34.42</u>	<u>41.81</u>	<u>74.11</u>	0.98	116,844
⑩	全ての記号	<u>18.48</u>	<u>11.71</u>	<u>14.23</u>	<u>100.00</u>	0.04	4,769
⑤①	日本語様相関数	<u>33.84</u>	<u>28.16</u>	<u>21.88</u>	<u>72.01</u>	0.19	22,654
⑤④	同値型グループ関数	<u>57.87</u>	<u>36.35</u>	<u>40.29</u>	<u>73.98</u>	0.88	104,922
⑤⑤	全ての関数	<u>30.55</u>	<u>25.89</u>	<u>19.81</u>	<u>72.14</u>	0.14	16,692
⑤⑥	全ての記号・関数	<u>8.90</u>	<u>8.21</u>	<u>6.54</u>	<u>100.00</u>	0.01	1,192

~~~~~は❶より値が低いとき, \_\_\_\_\_は❶より値が高いとき付与

表7: 評価パラメータの結果(文法・句レベル)

| 評価項目     |           | $R1$         | $N$           | $R2\_W\_all$ | $R2\_W\_match$ | $\eta_{R1}$ | 換算値 $X$ |
|----------|-----------|--------------|---------------|--------------|----------------|-------------|---------|
| 文型パターン辞書 |           |              |               |              |                |             |         |
| ❶        | 基準パターン    | 81.53        | 261.73        | 67.50        | 85.63          | 1.00        | 75,708  |
| ①        | 全ての離散記号   | <u>58.42</u> | <u>217.52</u> | <u>52.76</u> | <u>100.00</u>  | 0.16        | 12,113  |
| ②        | 文節境界記号    | <u>81.80</u> | <u>261.60</u> | <u>67.70</u> | <u>85.52</u>   | 1.02        | 77,222  |
| ⑥        | 全ての要素選択記号 | <u>80.78</u> | <u>226.73</u> | <u>66.33</u> | <u>84.99</u>   | 0.93        | 70,408  |
| ⑦        | 任意要素記号    | <u>81.33</u> | <u>251.57</u> | <u>67.02</u> | <u>85.26</u>   | 0.98        | 74,194  |
| ⑧        | 順序任意記号    | 81.53        | <u>261.68</u> | <u>67.50</u> | <u>85.62</u>   | 1.00        | 75,708  |
| ⑨        | 位置変更可能記号  | <u>81.46</u> | <u>247.64</u> | <u>67.26</u> | <u>85.42</u>   | 0.99        | 74,951  |
| ⑩        | 全ての記号     | <u>56.21</u> | <u>172.03</u> | <u>50.54</u> | <u>100.00</u>  | 0.14        | 10,599  |
| ⑤①       | 日本語様相関数   | <u>47.59</u> | <u>196.70</u> | <u>35.86</u> | <u>81.70</u>   | 0.07        | 5,300   |
| ⑤④       | 同値型グループ関数 | <u>78.85</u> | <u>260.80</u> | <u>65.29</u> | <u>85.90</u>   | 0.79        | 59,809  |
| ⑤⑤       | 全ての関数     | <u>44.01</u> | <u>187.42</u> | <u>33.93</u> | <u>83.92</u>   | 0.05        | 3,785   |
| ⑤⑥       | 全ての記号・関数  | <u>26.29</u> | <u>123.32</u> | <u>22.39</u> | <u>100.00</u>  | 0.01        | 757     |

表 8: 評価パラメータの結果 (文法・節レベル)

| 評価項目     |           | R1    | N       | R2_W_all | R2_W_match | $\eta_{R1}$ | 換算値 X  |
|----------|-----------|-------|---------|----------|------------|-------------|--------|
| 文型パターン辞書 |           |       |         |          |            |             |        |
| ❶        | 基準パターン    | 94.12 | 1049.10 | 87.08    | 93.51      | 1.00        | 22,766 |
| ①        | 全ての離散記号   | 78.69 | 254.64  | 74.23    | 100.00     | 0.20        | 4,553  |
| ②        | 文節境界記号    | 94.13 | 1049.76 | 87.11    | 93.52      | 1.00        | 22,766 |
| ⑥        | 全ての要素選択記号 | 93.98 | 903.37  | 86.98    | 93.44      | 0.98        | 22,311 |
| ⑦        | 任意要素記号    | 94.13 | 1038.54 | 87.11    | 93.49      | 1.00        | 22,766 |
| ⑧        | 順序任意記号    | 94.13 | 1049.74 | 87.12    | 93.51      | 1.00        | 22,766 |
| ⑨        | 位置変更可能記号  | 94.14 | 1048.06 | 87.12    | 93.46      | 1.00        | 22,766 |
| ⑩        | 全ての記号     | 78.35 | 219.04  | 73.92    | 100.00     | 0.19        | 4,326  |
| ⑤①       | 日本語様相関数   | 95.84 | 2554.13 | 88.53    | 93.75      | 1.36        | 30,962 |
| ⑤④       | 同値型グループ関数 | 94.39 | 1062.60 | 87.21    | 93.39      | 1.05        | 23,904 |
| ⑤⑤       | 全ての関数     | 95.77 | 2585.52 | 88.44    | 93.76      | 1.34        | 30,506 |
| ⑤⑥       | 全ての記号・関数  | 80.77 | 518.35  | 76.11    | 100.00     | 0.23        | 5,236  |

表 9: 評価パラメータの結果 (文法・全レベル)

| 評価項目     |           | R1    | N       | R2_W_all | R2_W_match | $\eta_{R1}$ | 換算値 X   |
|----------|-----------|-------|---------|----------|------------|-------------|---------|
| 文型パターン辞書 |           |       |         |          |            |             |         |
| ❶        | 基準パターン    | 95.90 | 1275.17 | 89.10    | 93.99      | 1.00        | 217,703 |
| ①        | 全ての離散記号   | 81.70 | 404.70  | 77.17    | 100.00     | 0.31        | 67,488  |
| ②        | 文節境界記号    | 95.92 | 1276.31 | 89.13    | 94.01      | 1.00        | 217,703 |
| ⑥        | 全ての要素選択記号 | 95.85 | 1096.64 | 88.98    | 93.86      | 0.99        | 215,526 |
| ⑦        | 任意要素記号    | 95.83 | 1253.52 | 89.02    | 93.94      | 0.99        | 215,526 |
| ⑧        | 順序任意記号    | 95.91 | 1275.59 | 89.14    | 93.99      | 1.00        | 217,703 |
| ⑨        | 位置変更可能記号  | 95.91 | 1260.55 | 89.12    | 93.96      | 1.00        | 217,703 |
| ⑩        | 全ての記号     | 81.09 | 333.56  | 76.59    | 100.00     | 0.30        | 65,311  |
| ⑤①       | 日本語様相関数   | 96.23 | 2650.96 | 89.26    | 94.07      | 1.05        | 228,588 |
| ⑤④       | 同値型グループ関数 | 95.82 | 1283.30 | 88.93    | 93.92      | 0.99        | 215,526 |
| ⑤⑤       | 全ての関数     | 96.14 | 2669.60 | 89.15    | 94.05      | 1.04        | 226,411 |
| ⑤⑥       | 全ての記号・関数  | 81.60 | 553.71  | 76.94    | 100.00     | 0.31        | 67,488  |

表 6~9 から、以下のことが分かる。

- (1) 記号では、離散記号が単語・句・節・全レベルで最も汎化効果があると分かった。離散記号を文型パターン辞書の記述に用いると、全レベルの場合では、文型パターン辞書を 3.23 倍に拡大したことが分かった (辞書 ❶ の  $\eta_{R1}$  ÷ 辞書 ① の  $\eta_{R1}$ )。
- (2) 関数では、日本語様相関数が単語・句レベルで最も汎化効果があると分かった。
- (3) 関数では、同値型グループ関数が節・全レベルで最も汎化効果があると分かった。同値型グループ関数を文型パターン辞書の記述に用いると、全レベルの場合文型パターン辞書を 1.01 倍に拡大したことが分かった (辞書 ❶ の  $\eta_{R1}$  ÷ 辞書 ⑤④ の  $\eta_{R1}$ )。

- (4) 全ての記号・関数を文型パターン辞書の記述に用いると、全レベルの場合文型パターン辞書を 3.23 倍に拡大したことが分かった（辞書 ① の  $\eta_{R1}$  ÷ 辞書 ⑥ の  $\eta_{R1}$ ）。

これらより、記号・関数が文型パターンの汎用性を高めていることを確認できた。

## 5 汎化作業コスト

### 5.1 汎化作業の関係

記号・関数を、単語・句・節レベルの文型パターンに導入する際、日英対訳文対における統語的・意味的な関係をアナリストが分析し、導入可能であるかどうかを検査している。その前後には実務的な作業が関わっている。基礎的な作業として、(a) ツール作成 (b) データ抽出 (c) 実作業（分析作業）(d) 作業結果の反映という段階がある (c) はツール上で作業する場合もあれば、紙の上での作業と入力作業が混在する場合もある。(d) は、中枢のデータベースに作業結果の形式を整えつつ登録する部分がある。

文型パターンの記述子ごとに参照する言語的情報の要点がある。具体的に基礎となる言語的情報を表 10 の大項目と小項目に示し、その (a) ~ (d) の作業コストを人日の単位で示す。

また、記述子ごとの作業では、この言語的情報を、部分的かつ混合して使用するため、作業工数の算出は単純ではないが、表 11 の「小計」の列の通りとなった（汎化作業コストについては 5.2 節で述べる）。

例えば、表 11 によると、離散記号の導入は、「日本語形態素の品詞等の情報 (2-1)」，および「日本語節間キーワード (3)」を基礎情報として使用し、ある程度自動的に文型パターンに導入していることが (d) で読み取れる。

表 10: 基礎となる言語的情報の作業工数

| 大項目            | 小項目                   | (a) | (b) | (c) | (d) | 小計  |
|----------------|-----------------------|-----|-----|-----|-----|-----|
| 1 日英対応付け作業     | 変数化, 補完主語, 英語時制等の対応付け | 40  | 20  | 620 | 20  | 700 |
| 2 日本語形態素修正     | 1 品詞, 単語・文節境界等の修正     | 30  | 10  | 60  | 5   | 105 |
|                | 2 標準表記の修正             |     | 5   | 20  | 5   | 30  |
| 3 日本語節間キーワード付与 |                       |     |     | 80  |     | 80  |

(単位 = 人日)

表 11: 記述子ごとの導入作業の工数と作業コスト

| 記述子類 | 記述子名     | 使用した基礎情報 | (a) | (b) | (c) | (d) | 小計  | 汎化作業コスト |
|------|----------|----------|-----|-----|-----|-----|-----|---------|
| 記号   | 離散記号     | 2-1, 3   |     |     |     | 40  | 40  | 225     |
|      | 文節境界記号   | 2-1, 3   |     |     |     | 10  | 10  | 195     |
|      | 要素選択記号   | 2        | 5   | 10  | 60  | 20  | 95  | 230     |
|      | 任意要素記号   | 1        |     |     |     |     | 0   | 920     |
|      | 順序任意記号   | 2-1      | 30  | 15  | 140 | 10  | 195 | 300     |
|      | 位置変更可能記号 | 2-1      | 30  | 15  | 180 | 10  | 235 | 340     |
| 関数   |          | 1, 2-1   |     |     |     | 100 | 100 | 1,125   |

(単位 = 人日)

## 5.2 記号・関数に対する汎化作業コスト

記号・関数に対する「汎化作業コスト」は、表 11 の「小計」と「使用した基礎情報（表 10 から算出）」の和で算出する。例えば、離散記号の汎化作業コストは、「小計」+「使用した基礎情報」=  $40 + (105 + 80) = 225$  人日となる。表 11 から、以下のことが分かる。

- (1) 汎化作業コストが最も低いのは、文節境界記号の 195 人日である。
- (2) 汎化作業コストが最も高いのは、関数の 1,125 人日である。

これらから、汎化作業コスト間で、最大 6 倍程度の差があることが分かった。



## 6 総合評価

第4章で求めた「汎用性の評価」と第5章で求めた「汎化作業コスト」から、1人日あたりの文型パターン作成数に換算した総合評価を行なう。算出方法は、4.1.1節の基準パターン辞書の文型パターン数と表6~9の換算値  $X$  との差を求め、表11の汎化作業コストで割ることで算出する。ただし、それぞれの汎化レベルごとに総合評価を行なうため、汎化作業コストは、4.1.1節の汎化レベルごとの基準パターン辞書の割合を汎化作業コストに掛けた値を使用する。

例えば、離散記号の場合は、「文型パターン作成数」÷「汎化作業コスト」=  $(119,229 - 8,346) \div (225 \times 0.55) = 896$  となる（単語レベルの場合）。表12~15に、1人日あたりの文型パターン作成数が多い順の総合評価結果を示す。

表 12: 総合評価結果（単語レベル）

| 順位 | 記述子名（類）  | 1人日あたりの文型パターン作成数<br>(文型パターン作成数 / 作業コスト) |                    |
|----|----------|-----------------------------------------|--------------------|
| 1  | 離散記号     | 896                                     | (110,883 / 123.75) |
| 2  | 関数       | 166                                     | (102,537 / 618.75) |
| 3  | 任意要素記号   | 66                                      | (33,384 / 506.00)  |
| 4  | 要素選択記号   | 47                                      | (5,961 / 126.50)   |
| 5  | 位置変更可能記号 | 13                                      | (2,385 / 187.00)   |
| 6  | 順序任意記号   | 7                                       | (1,192 / 165.00)   |

表 13: 総合評価結果（句レベル）

| 順位 | 記述子名（類）  | 1人日あたりの文型パターン作成数<br>(文型パターン作成数 / 作業コスト) |                   |
|----|----------|-----------------------------------------|-------------------|
| 1  | 離散記号     | 808                                     | (63,595 / 78.75)  |
| 2  | 関数       | 183                                     | (71,923 / 393.75) |
| 3  | 要素選択記号   | 66                                      | (5,300 / 80.50)   |
| 4  | 位置変更可能記号 | 6                                       | (757 / 119.00)    |
| 5  | 任意要素記号   | 5                                       | (1,514 / 322.00)  |
| 6  | 順序任意記号   | 0                                       | (0 / 102.00)      |

表 14: 総合評価結果 (節レベル)

| 順位 | 記述子名 (類) | 1人日あたりの文型パターン作成数<br>(文型パターン作成数 / 作業コスト) |                    |
|----|----------|-----------------------------------------|--------------------|
| 1  | 離散記号     | 809                                     | (18,213 / 22.50)   |
| 2  | 要素選択記号   | 20                                      | (455 / 23.00)      |
| 3  | 順序任意記号   | 0                                       | (0 / 30.00)        |
| 3  | 位置変更可能記号 | 0                                       | (0 / 34.00)        |
| 3  | 任意要素記号   | 0                                       | (0 / 92.00)        |
| 6  | 関数       | -91                                     | (-10,245 / 112.50) |

表 15: 総合評価結果 (全レベル)

| 順位 | 記述子名 (類) | 1人日あたりの文型パターン作成数<br>(文型パターン作成数 / 作業コスト) |                    |
|----|----------|-----------------------------------------|--------------------|
| 1  | 離散記号     | 668                                     | (150,215 / 225.00) |
| 2  | 要素選択記号   | 9                                       | (2,177 / 230.00)   |
| 3  | 任意要素記号   | 2                                       | (2,177 / 920.00)   |
| 4  | 位置変更可能記号 | 0                                       | (0 / 340)          |
| 4  | 順序任意記号   | 0                                       | (0 / 300)          |
| 6  | 関数       | -8                                      | (-8708 / 1,125.00) |

表 12 ~ 15 から、以下のことが分かる。

- (1) 離散記号で、単語・句・節・全レベルにおいて、総合評価が最も良く、少量の作業コストで、文型パターンが大量に作成できる。
- (2) 関数は、種類数が多く作業コストがかかる。単語・句レベルでは汎化の効果が高く、導入は妥当であると言える。しかし、節・全レベルでは汎化の効果が低下して、文型パターンが失われる結果となった。

## 7 考察

### 7.1 意味的排他性の評価

記号・関数を用いた文型パターンが、意味的に適切な文型パターンであるか、また、それに対応する英語文型パターンが英訳の生成に問題なく使用できるかは大変重要である。そこで、入力文をテスト用入力文からランダムに30文選び、記号・関数の中で最も効果があった離散記号、日本語様相関数を削除した文法・単語レベル文型パターン辞書を対象に意味的排他性に関する評価を行った。

#### 7.1.1 意味的排他性の評価パラメータ

本稿では、意味的排他性を評価するパラメータとして、「適合文型意味正解率  $P1$ 」、「適合文型意味含有率  $P2$ 」を使用する [3]。

〈適合文型意味正解率  $P1$ 〉

$P1$  は「入力文に適合した文型パターンが、意味的に適切な文型パターンである確率」を表し、以下の式で定義される。

$$P1 = n/N \quad (7)$$

$N$ ：適合した文型パターンの数

$n$ ：意味的に適切な文型パターンの数

〈適合文型意味含有率  $P2$ 〉

$P2$  は「入力文に対して、意味的に適切な文型パターンが1つ以上含まれる確率」を表し、以下の式で定義される。

$$P2 = m/k \quad (8)$$

$k$ ：適合した文型パターンがある文の数

$m$ ：意味的に適切な文型パターンが1つ以上存在する文の数

〈意味的被覆率  $R1 \times P2$ 〉

文型パターン方式に期待される効果の最大値を表す。

#### 7.1.2 排他性の評価結果

意味的排他性の評価結果以下に示す。

- 基準パターン

$R1 : 63.33 \% , P1 : 5.87 \% , P2 : 47.37 \% , R1 \times P2 : 29.99 \%$

- 離散記号

$R1 : 33.33 \% , P1 : 66.67 \% , P2 : 80.00 \% , R1 \times P2 : 26.66 \%$

- 日本語様相関数

$R1 : 40.00 \% , P1 : 12.15 \% , P2 : 41.66 \% , R1 \times P2 : 16.66 \%$

これより、 $P1, P2$  が最も高いのは離散記号だと分かった。また、意味的被覆率を表す  $R1 \times P2$  が最も高いのは基準パターンとなり、離散記号、日本語様相関数を用いたときのほうが、意味的被覆率が向上することが確認できた。

### 7.1.3 文型パターン数の換算値と排他性の評価結果

表6から、離散記号と日本語様相関数を削除したときの文型パターン数の換算値はそれぞれ8,346パターンと22,654パターンとなる。この文型パターン数を基準パターン辞書からランダムに5回選択した文型パターン辞書と30文の意味的排他性の平均評価結果を以下に示す。

- 基準パターン (8,346パターン)

$R1 : 15.33 \% , P1 : 5.71 \% , P2 : 34.78 \% , R1 \times P2 : 5.33 \%$

- 基準パターン (22,654パターン)

$R1 : 30.67 \% , P1 : 5.68 \% , P2 : 34.78 \% , R1 \times P2 : 10.67 \%$

7.1.2節の評価結果と文型パターン数の換算値での排他性の評価結果から、文型パターン数の換算値では、排他性の評価は行なえないと考えられる。

### 7.1.4 離散記号の効果があつた例

離散記号を用いることによって、意味的に正しい英訳が出力できた例を示す。

- 入力文：彼の家族はみんな風変わりな連中だった。

- (日) 適合パターン：/y #1[/tcfk N2 の] /k N3 は /cf AJV4^rentai ! N5.#da.kako。

- N2 = 彼

- N3 = 家族
- /cf = みんな
- AJV4 = 風変わりな
- N5 = 連中

- (英) 適合パターン : #1[N2<sup>poss</sup>] N3 be<sup>past</sup> AJ4 N5.
- 出力英文 : His family were queer lot .

この例文では離散記号「/cf」が「みんな」に適合する。出力英文には「みんな」の意味が反映されないが、それ以外は質の高い英文が出力されている。この場合、離散記号を使わないと上記の例文は文型パターンと適合しなくなるため、離散記号の使用効果が確認できた。ただし、離散記号と適合した入力文の要素が、英語翻訳の際の重要な語になる場合、もしくは、入力文のほとんどが離散記号に適合してしまう場合は、意味的に正しくない英訳になった。

## 8 おわりに

本稿では、文型パターンの汎用性を高めるための文型パターンの記述子である「記号」と「関数」について、その汎化効果の評価、および、汎化作業コストを求めて、総合評価を行った。

全（単語・句・節）レベルの文型パターン汎化効果の評価の結果、記号で最も汎化効果があるのは、離散記号と分かった。離散記号を文型パターン辞書の記述に用いると、文型パターン辞書を 3.23 倍に拡大したことが分かった。関数で最も汎化効果があるのは、同値型グループ関数と分かった。同値型グループ関数を文型パターン辞書の記述に用いると、文型パターン数が 1.01 倍に拡大したことが分かった。全ての記号・関数を文型パターン辞書の記述に用いると、文型パターン数が 3.23 倍に拡大したことが分かった。

作業コストが最も低いのは、文節境界記号の 195 人日で、最も高いのは、関数の 1,125 日だと分かった。

全レベルの文型パターンの汎化効果の評価結果と作業コストから、総合評価が最も良いのは、離散記号で、1 人日あたり 668 パターン作成出来ることが分かった。離散記号は、単語・句・節・全レベルにおいて少量の作業コストで、文型パターンが大量に作成出来ることが分かった。

これらから、記号・関数は、文型パターンの汎用性を高めていることを確認出来た。また、一般のパターン型知識ベースの開発における、コストの判断に有効な参考値を見出すことが出来た。

## 謝辞

本研究・本論文作成に際して、多大なる検討と種々の御助言をしていただきました鳥取大学工学部知能情報工学科計算機C工学講座池原研究室の池原悟教授，村上仁一助教授に心からお礼を申し上げます。

また，終始に渡り御指導いただきました徳久雅人助手に心から感謝致します。

そして，本論文について多くの御意見，御助言を頂きました菅原一孔教授に深く感謝致します。

その他，本研究に使用させて頂いた本の著者の方々，および様々な場面で御助力・励ましの御言葉をかけてくださった計算機C工学講座池原研究室の皆様にも深く感謝の意を表します。

## 発表文献

- [1] 金澤佑哉, 徳久雅人, 村上仁一, 池原悟:記号と関数による文型パターン汎化効果の総合評価, 言語処理学会第 13 回年次大会, 2007(発表予定).

## 参考文献

- [1] 池原悟, 阿部さつき, 徳久雅人, 村上仁一:非線形な表現構造に着目した重文と複文の日英文型パターン化, 自然言語処理, 11(3), pp.69-95, 2004.
- [2] 池原悟, 徳久雅人, 村上仁一, 佐良木昌, 池田尚志, 宮崎正弘:非線形な重文複文の表現に対する文型パターン辞書の開発, 情報処理学会研究報告, 2005-NL-170, pp.157-164, 2005.
- [3] 池原悟, 徳久雅人, 竹内(村本)奈央, 村上仁一:日本語重文・複文を対象とした文法レベル文型パターンの被覆率特性, 自然言語処理, 11(4), pp.147-178, 2004.
- [4] 遠藤久美子, 徳久雅人, 村上仁一, 池原悟:文型パターンにおける任意要素の記述方法とその効果, 言語処理学会第 11 回年次大会, pp.368-371, 2005.
- [5] 金澤佑哉, 徳久雅人, 村上仁一, 池原悟:文型パターンにおける時制・相・様相表現の汎化とその効果, 言語処理学会第 11 回年次大会, pp.29-32, 2005.
- [6] 徳久雅人, 村上仁一, 池原悟:重文・複文文型パターン辞書からの構造照合型パターン検索, 情報処理学会研究報告, 自然言語処理, 2006-NL-176, pp.9-16, 2006.