概要

日英機械翻訳において，文型パターンを用いて翻訳する方法がある．その中でも、等価的類推思考の原理に基づく機械翻訳方式では，翻訳対象となる日英両言語で言語表現を「文型パターン」によって記述しておき，意味的に等価な文型パターンを対応付けることで，意味の失われない解析・生成を実現しようとしている．

文型パターンは,「字面」,「変数」,「関数」，および,「記号」で構成される．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．逆にそうでないものは，パターンの構成要素にならない．そして，パターンマッチングにより入力文の解析を行うものである．文型パターンの汎用性を高めるためには，異なる時制・相・様相への適合や，日本語に顕著な語順の自由度を高める等の，記述要素の効果的な汎化が課題となっている．

そこで，本稿では，日本語パターンにおける「パターンに時制・相・様相の情報を与える関数」，および,「要素の位置変更可能の指定」についての効果を評価パラメータを用いて，汎化の効果について，定量的に評価する．

具体的な方法は，次の通りである．まず，文型パターン辞書に収録されている文法・単語レベルの文型パターン辞書 (122,619 パターン) を「基準パターン辞書」とする．そして，基準パターン辞書に対して「時制・相・様相関数の汎化」，および,「位置変更可能記号の無効化」を行い，それぞれに対し新たな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 123,016 文と基準パターン辞書，および，対象パターン辞書を文型パターンパーサを用いて照合を行う．そして，照合結果から，評価パラメータとして「文型パターン拡大率 η」（対象パターン辞書の規模が基準パターン辞書の規模に換算して，何倍に相当するか）を用いて文型パターン辞書の汎化及び無効化の効果を定量的に評価し，汎化の効果を確認する．

評価の結果，基準パターン辞書と比べ時制を汎化すると 1.36〜1.53 倍，相・様相を汎化すると 1.40〜1.49 倍，位置変更可能記号を有効化すると無効化した場合に比べ 1.02〜1.05 倍規模のパターン辞書が作成でき,「時制・相・様相を表す関数」，および,「要素の位置変更可能の指定」をすることは，パターンの汎用性を向上するために有効だとわかった．

目 次

表 目 次

図 目 次

1　はじめに

等価的類推思考の原理に基づく言語の等価交換を実現するために，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,「字面」,「変数」,「関数」，および,「記号」で構成される．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．逆にそうでないものは，パターンの構成要素にならない．そして，パターンマッチングにより入力文を解析する．文型パターンの汎用性を高めるために，記述要素の効果的な汎化が課題となっている．

[1] で構築されているパターン辞書は，10 万オーダーのパターンで構成されている．このパターン辞書では,「パターンに時制・相・様相の情報を与える関数」が入力文と一致しなければパターンが適合しない．今後，パターンの汎用性を向上させるには，これらの制約を弱めること，日本語に顕著な語順の自由度を高めることなどが考えられる．パターン辞書の全面的な改良には大きなコストがかかるため，様々考えられる改良項目の中でも，効果の見込まれる項目を行う必要がある．

そこで，本稿では「時制・相・様相を表す関数の汎化」，および,「要素の位置変更可能の指定」についての効果を,「文型パターン拡大率 η」[2] を用いて，汎化の効果について，定量的に評価する．

本稿の構成は以下の通りである．第 2 章で被覆率調査における評価パラメータの種類と評価パラメータの測定方法について述べる．第 3 章で時制・相・様相の汎化と位置変更可能記号の無効化について述べる．第 4 章で汎化・無効化の効果の実験について，実験の目的，実験の方法，実験の結果について述べる．第 5 章で汎化・無効化の効果について実験の結果を基に考察を行う．第 6 章でまとめを述べ，今後の課題を提案する．

2　被覆率調査における評価パラメータ

2.1　文型パターン

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る．原文は，単語レベル，句レベル，節レベルの 3 レベルにパターン化され，各レベルに応じた粒度でアライメントがとれた部分は，線形要素として変数化される．逆に，変数化すると対訳の訳出が困難になる部分は変数化せず，非線形要素として字面，あるいは関数の形式で残される．

文型パターンの具体例を以下に示す．

- 日本語原文=彼は不幸にも金をなくした。
- 英語原文=He had the misfortune to lose his money.
- 日本語パターン $=N1$ は不幸にも $N2$ を $V3.kako$。
- 英語パターン $=N1$ had the misfortune to $V3$ $N1$.poss $N2$.

N は名詞，V は動詞，$.kako$ は時制を表している．英語パターンは日本語パターンを元に，英語原文を変数化して作成されている．

本稿では，[1] の文型パターンのうち文法・単語レベルの文型パターン辞書（122,619 パターン）を「基準パターン辞書」とする．この辞書を対象に汎化・無効化を行い，新たなパターン辞書「対象パターン辞書」を作成する．

2.2　評価パラメータの種類

入力日本文の翻訳に使用できる文型パターンは，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．文型照合プログラムは，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，適合文型として抽出するので，適合文型は，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，適合文型とはいえ，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，適合文型について，それが入力文の何%をカバーしているかが問題となる．

そこで本稿では，[2]，[4] 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．

(1) 文型再現率 $R1$

$R1$ は「全入力文のうち，適合文型パターンが存在した入力文の割合」を表し，以下の式で定義される．

$$R1 = M/I \tag{1}$$

I：テスト用入力文の数

M：「自己文型パターン」以外の適合文型パターンが 1 つ以上存在した入力文の数

(2) 文型一致率 $R2$

$R2$ は「入力文の文字単位に見た再現率」を表すもので，以下の式で定義される．

$$R2 = \sum_{j=1}^{k} M_j / \sum_{i=1}^{I} N_i \tag{2}$$

Ni：i 番目の入力文の文字数

k：自己文型パターン以外に適合文型パターンがある入力文の数

Mj：「最尤文型パターン」と一致する入力文の文字数

但し,「最尤文型パターン」とは,「自己文型パターン」以外の適合文型パターンのうち,最も広範囲に入力文と一致する文型パターンを言う.

(3) 完全一致平均適合パターン数 $N1$

$N1$ は「入力文に対する完全一致適合文型パターン数の平均値」を表し，以下の式で定義される．

$$N1 = P_{N1}/I \tag{3}$$

P_{N1}：完全一致適合文型パターン数

(4) 部分一致平均適合パターン数 $N2$

$N2$ は「入力文に対する部分一致適合文型パターン数の平均値」を表し，以下の式で定義される．

$$N2 = P_{N2}/I \tag{4}$$

P_{N2}：部分一致適合文型パターン数

(5) 平均適合パターン数 N

N は「入力文に対する適合文型パターン数の平均値」を表し，以下の式で定義される．

$$N = P/I = N1 + N2 \tag{5}$$

P：適合文型パターン数

(6) 文型パターン拡大率 η

η は「評価対象の対象パターン辞書が基準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．

$$\eta = X/B \tag{6}$$

B：基準パターン辞書の文型パターン数

X：対象パターン辞書の文型パターン数の換算値

2.3 評価パラメータの測定方法

η を用いるには，基準パターン辞書の文型パターン数と対象パターン辞書の文型パターン数の換算値が必要となる．本稿では「文型再現率 $R1$」からみた「文型パターン拡大率 η_{R1}」と「平均適合パターン数 N」からみた「文型パターン拡大率 η_N」を用いる．以下に，η_{R1} と η_N の概略を [2] より説明する．

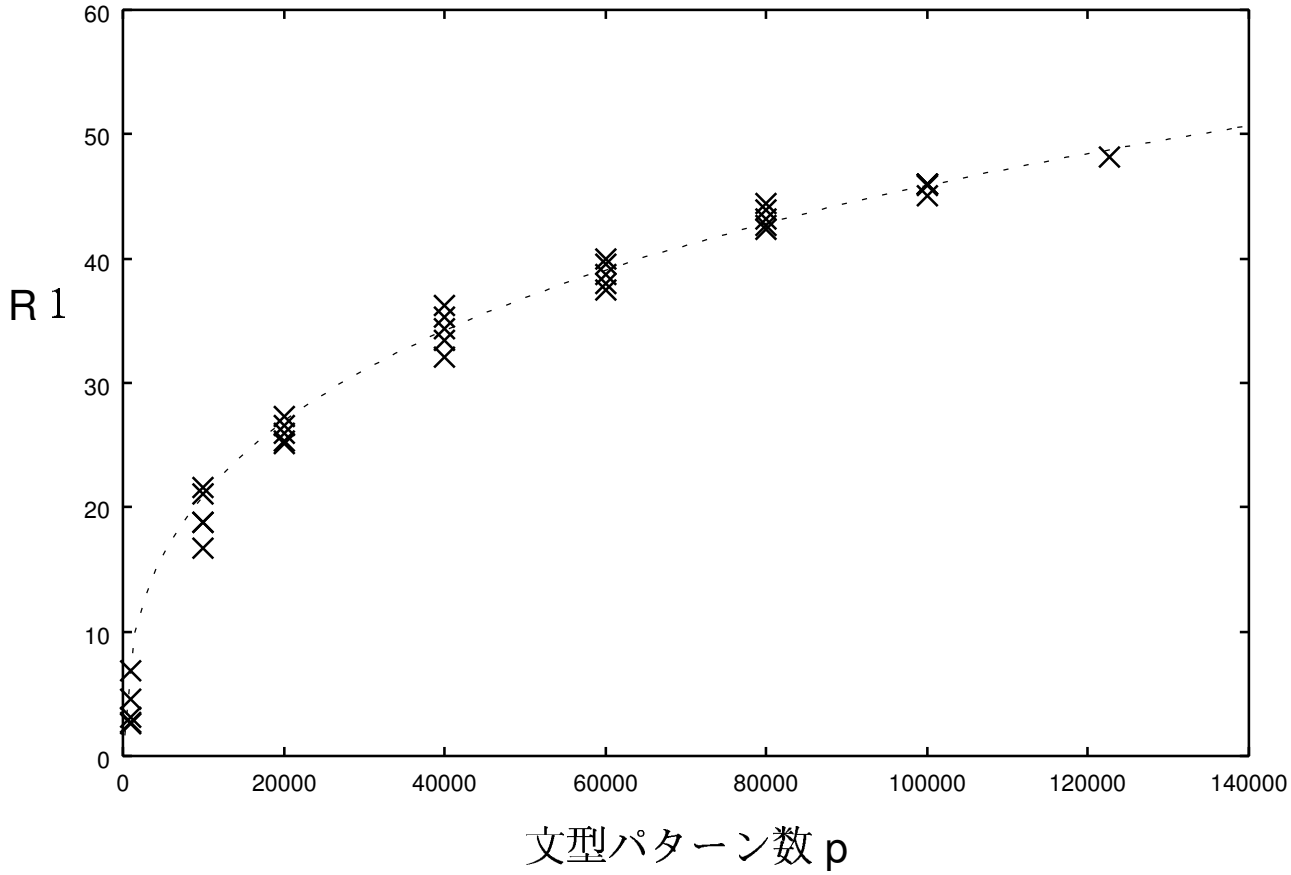

図 1: $R1$ と文型パターン数の関係図

η_{R1} の推定方法

基準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 1 で示す．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，(7) 式で近似する．近似曲線を点線で示す．

$$R1 = (1 - exp(-\lambda_1 (p_{R1})^{\lambda_2})) * 100.0(\%) \tag{7}$$

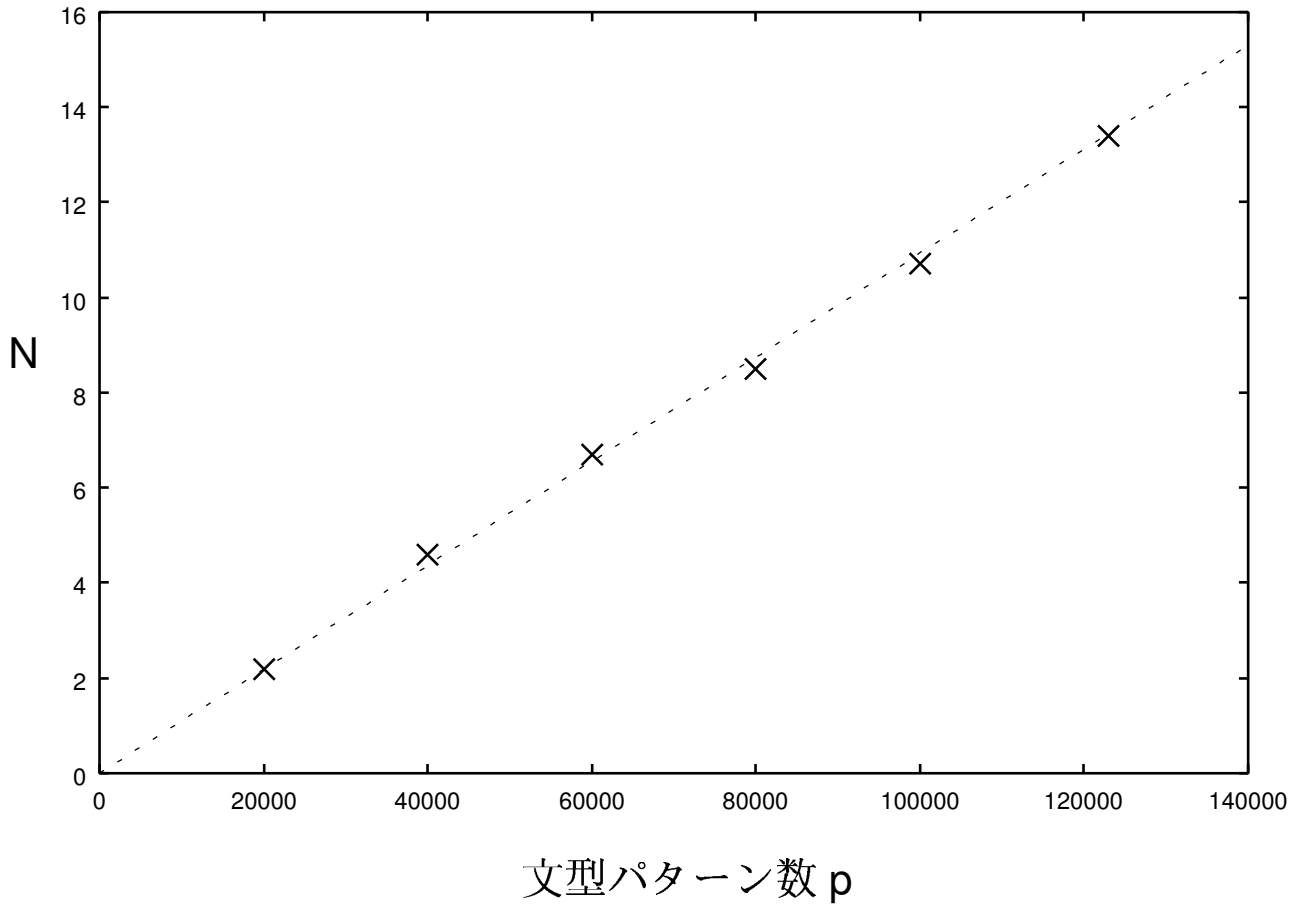

図 2: N と文型パターン数の関係図

η_N の推定方法

同じく基準パターン辞書を用いた実験から，平均適合パターン数 N と文型パターン数 p の関係を，図 2 に示す．図中の縦軸は，平均適合パターン数 N 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 N の実測値を示す．

線形回帰分析より，平均適合パターン数 N の文型パターン数 p に対する特性を，(8) 式で近似する．近似線を図中に点線で示す．

$$N = \lambda_3 p_N \tag{8}$$

これより，基準パターン辞書の実測値より，λ_1，λ_2，λ_3 を求め，対象パターン辞書の実測値 $R1$，N より，p_{R1} および p_N を求める．そして，(6) 式にそれぞれ代入することで，η_{R1} および η_N を求める．

2.4　研究の目的

汎化の対象は，文型パターンの述部に記述された時制・相・様相である．これらの表現はすでに関数化されており，39 種類の関数が定義されている．

実際には，これらの表現を汎化すると，文全体の意味が異なってしまうことがある．しかし，本稿では，汎化の効果の上限を調べ，その効果を確認することがねらいなので, 以下のアプローチをとる．

(1) 「汎化の規則が発見でき，機械的な適用が可能なものを汎化する」

(2) 「文型パターン内での使用頻度からみて，汎化の効果が大きいものを汎化する」

無効化の対象は，文型パターンに記述された位置変更可能記号である．これはすでに，パターンに記述済みである．

本稿では，位置変更可能記号の有る無しの効果を調べ，位置変更可能記号の有効性を確認することがねらいなので，以下のアプローチをとる．

(1) 「文型パターンに記述されている位置変更可能記号を無効にする」

これらのアプローチによって作成されたパターン辞書（対象パターン辞書を）と基準パターン辞書を，上記の評価パラメータを用い，汎化・無効化の効果を定量的に評価し，効果を確認する．

3　文型パターンにおける時制・相・様相の汎化と位置変更可能記号の無効化

3.1　時制関数の汎化の実施

3.1.1　時制関数の汎化の方法

時制関数とはパターンに時制の情報を与える関数であり，過去関数「*.kako*」と未来関数「*.darou*」の 2 種類がある．時制関数は文型パターンの述部に記述され，そのパターンに〈過去〉または〈未来〉の時制を与えている．時制関数が文型パターンの述部に記述されていないパターンは〈現在〉の時制であることが多い．

異なる時制への適合について，その汎化の効果を得るため，過去関数と未来関数を統合し，両時制に適合する新たな関数である過去未来関数「*.kakodarou*」を定義する．そして，過去未来関数に任意記号（照合ではその要素が入力日本文にあってもなくてもよいことを示す記号，[…] で表記する）を付けることにより現在時制にも適合するようにする．本稿ではこの関数を「自由時制関数」と呼ぶ．

自由時制関数をパターンに付加する方法は 2 つある．以下に手順を示す．

方法 1

手順 (1) 時制関数がパターンに記述されているに記述されているかを調べ，記述されている場合，

手順 (2) 時制関数を，自由時制関数に置換する．

方法 2

手順 (1) 時制関数がパターンに記述されているに記述されているかを調べ，記述されていない場合，

手順 (2) 基準パターン辞書から，時制関数と関数が組み合わさって記述されている述部の中で，時制関数が記述されている位置を調べる（表 1）．そして，[3] と表 1 より，パターンの述部に自由時制関数を挿入する位置を決める（表 2）．

手順 (3) 表 2 より，パターンの述部に自由時制関数を挿入する．

こうして，時制関数を汎化したパターン辞書（自由時制パターン辞書）を作成する．

3.1.2 時制関数の汎化の例

以下に時制関数の汎化している様子を示す.

方法 1

手順 (1) 時制関数がパターンに記述されているに記述されているかを調べ，記述されている場合

$WJ224583\text{-}01$: /y < /tk $N1$ は> /tcfk $N2$ /f $V3.hitei$ $\boxed{.darou}$ と /cf $V4$ $\boxed{.kako}$。

手順 (2) 時制関数を，自由時制関数に置換する.

$WJ224583\text{-}01$: /y < /tk $N1$ は> /tcfk $N2$/f$V3.hitei$ $\boxed{\#1[.kakodarou]}$ と /cf$V4$ $\boxed{\#2[.kakodarou]}$。

方法 2

手順 (1) 時制関数がパターンに記述されているに記述されているかを調べ，記述されていない場合

$WJ204615\text{-}01$: /y あらゆる /tk $N1$ の /k 背後に /tcfk $N2$ が /cf $V3.teiru$ と< /cf $N4$ は> /cf $V5.teiru.rashii$。

手順 (2) 基準パターン辞書から，時制関数と関数が組み合わさって記述されている述部の中で，時制関数が記述されている位置を調べる．そして，[3] と表 2 より，パターンの述部に自由時制関数を挿入する位置を決める.

$WJ204615\text{-}01$: /y あらゆる /tk $N1$ の /k 背後に /tcfk $N2$ が/cf$V3.teiru_{\square}$ と< /cf $N4$ は> /cf$V5.teiru_{\square}.rashii_{\square}$。

手順 (3) 表 2 より，パターンの述部に自由時制関数を挿入する.

$WJ204615\text{-}01$: /y あらゆる /tk $N1$ の /k 背後に /tcfk $N2$ が /cf $V3.teiru$ $\boxed{\#1[.kakodarou]}$ と< /cf $N4$ は> /cf $V5.teiru$ $\boxed{\#2[.kakodarou]}$ $.rashii$ $\boxed{\#3[.kakodarou]}$。

表 1: 時制 (*zisei*=過去関数・未来関数) 関数と関数が組み合わさっているパターンとその総数

数	関数	数	関数	数	関数
6,709	.*zisei*^rentai	5	.souda.joutaihenka.*zisei*	1	.temiru.yotei.*zisei*
1,886	.reru.*zisei*	5	.sase.teinei.*zisei*	1	.temiru.utosuru.*zisei*
1,765	.teiru.*zisei*	5	.reru.tekuru.teinei.*zisei*	1	.temiru.gimu.*zisei*
1,193	.hitei.*zisei*	5	.noda.*zisei*	1	.tekuru.tekureru.*zisei*
1,020	.*zisei*^katei	5	.joutaihenka.teshimau.*zisei*	1	.tekuru.rareru.teinei.*zisei*
749	.teinei.*zisei*	5	.joutaihenka.teinei.*zisei*	1	.tekuru.rareru.hitei.*zisei*
611	.tekuru.*zisei*	5	.hitei.yotei.*zisei*	1	.teiru.you.*zisei*
468	.rareru.*zisei*	5	.teoku.teinei.*zisei*	1	.teiru.tekureru.*zisei*
468	.*zisei*.you	5	.teiru.suitei.*zisei*	1	.teiru.teinei.hitei.teinei.*zisei*
283	^sase.*zisei*	4	.teiku.teinei.*zisei*	1	.teiku.tai.*zisei*
283	.sase.*zisei*	4	.reru.teiku.*zisei*	1	.tai.hitei.*zisei*
241	.tekureru.*zisei*	4	.kaishi.teinei.*zisei*	1	.sugiru.teiru.*zisei*
198	.*zisei*.noda	4	.joutaihenka.tekuru.*zisei*	1	.sase.teshimau.*zisei*
181	.gimu.*zisei*	4	.teyaru.hitei.*zisei*	1	.sase.tekureru.teinei.*zisei*
161	.teiku.*zisei*	4	.teoku.gimu.*zisei*	1	.sase.tekureru.hitei.*zisei*
124	.reru.teiru.*zisei*	3	.teyoi.*zisei*	1	.sase.rareru.teiru.*zisei*
120	.joutaihenka.*zisei*	3	.teiku.reru.*zisei*	1	.sase.rareru.teinei.*zisei*
116	.kaishi.*zisei*	3	.suitei.*zisei*	1	.sase.hitei.*zisei*
104	.temiru.*zisei*	3	.sugiru.teiku.*zisei*	1	.sase.gimu.*zisei*
95	.utosuru.*zisei*	3	.sase.utosuru.*zisei*	1	.reru.teshimau.teinei.*zisei*
83	.rareru.hitei.*zisei*	3	.rareru.teinei.hitei.teinei.*zisei*	1	.reru.teshimau.*zisei*
79	.teyaru.*zisei*	3	.*zisei*.noda .*zisei*	1	.reru.suitei.*zisei*
69	.*zisei*.suitei	3	.gimu^rentai.*zisei*	1	.reru.sugiru.*zisei*
62	.teiru.hitei.*zisei*	3	^sugiru.teiku.*zisei*	1	.teyaru.tai.*zisei*
61	.teiru.teinei.*zisei*	3	^sase.utosuru.*zisei*	1	.rareru.teiru.teinei.*zisei*
57	.reru.teinei.*zisei*	2	^sase.teiru.*zisei*	1	.rareru.teiku.*zisei*
47	.teinei.hitei.teinei.*zisei*	2	^sase.tearu.*zisei*	1	.rareru.kaishi.*zisei*
44	.rareru.teiru.*zisei*	2	.yotei.teinei.*zisei*	1	.nisuru.teoku.*zisei*
43	.*zisei*.suitei^rentai	2	.teyaru.tekuru.*zisei*	1	.nisuru.tekuru.*zisei*
40	.reru.tekuru.*zisei*	2	.teoku.tekureru.*zisei*	1	.nisuru.teiru.*zisei*
39	.yotei.*zisei*	2	.tekuru.teiru.*zisei*	1	.teyaru.gimu.*zisei*
37	.teshimau.teinei.*zisei*	2	.tekuru.rareru.*zisei*	1	.teshimau.souda.*zisei*
35	.teoku.*zisei*	2	.tekuru.*zisei*.suitei.*zisei*	1	.kaishi.teshimau.teiru.*zisei*
33	.tai.*zisei*	2	.tekureru.teinei.hitei.teinei.*zisei*	1	.joutaihenka.tekuru.teinei.*zisei*
32	.reru.hitei.*zisei*	2	.teiru.noda.*zisei*	1	.joutaihenka.kaishi.*zisei*
29	.tekuru.teinei.*zisei*	2	.teiru.gimu.*zisei*	1	.joutaihenka.hitei.*zisei*
27	.tearu.*zisei*	2	.sugiru.teinei.*zisei*	1	.hitei.suitei.*zisei*
26	.*zisei*.hougayoi	2	.sase.tekuru.*zisei*	1	.hitei.joutaihenka.teshimau.*zisei*
25	^sugiru.*zisei*	2	.sase.teiru.*zisei*	1	.hitei.joutaihenka.teinei.*zisei*
25	.sugiru.*zisei*	2	.sase.tearu.*zisei*	1	^sase.teshimau.*zisei*
24	^sase.rareru.*zisei*	2	.reru.teshimau.teiru.*zisei*	1	^sugiru.teiru.*zisei*
24	.sase.rareru.*zisei*	2	.reru.teiru.teinei.*zisei*	1	.teoku.reru.*zisei*
22	.*zisei*.rashii	2	.reru.teinei.hitei.teinei.*zisei*	1	.teoku.hitei.*zisei*
18	.tekureru.hitei.*zisei*	2	.rareru.teiru.hitei.*zisei*		
16	.reru.gimu.*zisei*	2	.*zisei*^rentai.noda		
14	.*zisei*.souda	2	.*zisei*.you^rentai		
13	.teshimau.teiru.*zisei*	2	.*zisei*.rashii.*zisei*		
12	.temiru.teinei.*zisei*	2	.*zisei*.hitei.*zisei*		
11	.souda.*zisei*	2	.*zisei*.hitei		
10	.tekuru.hitei.*zisei*	2	.joutaihenka.teiru.*zisei*		
10	.nisuru.*zisei*	2	.gimu.tearu.*zisei*		
9	.reru.teiru.hitei.*zisei*	2	^sugiru.teinei.*zisei*		
9	.rareru.teinei.*zisei*	2	^sase.tekuru.*zisei*		
9	.*zisei*.teinei.you	1	^sase.tekureru.teinei.*zisei*		
8	.utosuru.hitei.*zisei*	1	^sase.tekureru.hitei.*zisei*		
8	.rareru.tekuru.*zisei*	1	^sase.rareru.teiru.*zisei*		
8	.*zisei*.suitei .*zisei*	1	^sase.rareru.teinei.*zisei*		
8	.hitei.joutaihenka.*zisei*	1	^sase.hitei.*zisei*		
7	.utosuru.teiru.*zisei*	1	^sase.gimu.*zisei*		
7	.tekureru.teinei.*zisei*	1	.utosuru.temiru.teinei.*zisei*		
7	.tekudasai.teinei.*zisei*	1	.utosuru.tekuru.*zisei*		
7	.rareru.gimu.*zisei*	1	.utosuru.teiru.teinei.*zisei*		
7	.kaishi.teiru.*zisei*	1	.utosuru.teinei.*zisei*		
6	^sase.tekureru.*zisei*	1	.teyaru.temiru.*zisei*		
6	.teiru.rareru.hitei.*zisei*	1	.teyaru.teiru.*zisei*		
6	.sase.tekureru.*zisei*	1	.*zisei*.rashii^rentai		
6	.*zisei*.noda^katei	1	.*zisei*^rentai.suite		
5	^sase.teinei.*zisei*	1	.reru.*zisei*.noda .*zisei*		

表 2: 自由時制関数を挿入する場所

関数	自由時制関数の挿入場所
ˆrentai	*ˆrentai* 関数の前
ˆkatei	*ˆkatei* 関数の前
.you	*.you* 関数の前と相・様相関数の後ろ
.noda	*.noda* 関数の前と相・様相関数の後ろ
.suitei	*.suitei* 関数の前と相・様相関数の後ろ
.hougayoi	*.hougayoi* 関数の前と相・様相関数の後ろ
.rashii	*.rashii* 関数の前と相・様相関数の後ろ
.souda	*.souda* 関数の前と相・様相関数の後ろ
その他の相・様相関数	相・様相関数の後ろ

3.2 相・様相関数の汎化の実施

3.2.1 相・様相関数の汎化の方法

相・様相関数とはパターンに相および様相の情報を与える関数であり，37 種類が定義されている．相・様相関数も文型パターンの述部に記述され，そのパターンに相，様相の情報を与えている．

異なる相，様相への適合ついて，その汎化の効果を得るため，アプローチ (2) に沿って，パターン辞書中の相・様相関数の使用頻度を調べ（表 3），頻度が高い関数について，汎化を行う．表 3 より使用頻度が 1,000 以上の関数は 10 種類あり，その相・様相関数に対し以下の手順で汎化を行う．

手順 (1) 相・様相関数の中で意味的に包含関係をもつ他の相・様相関数が存在するか調べる．存在する場合は，手順 (2) を行い，存在しない場合は，手順 (3) を行う．

手順 (2) 相・様相関数の中で意味的に包含関係のもつ相・様相関数を統合し新たな関数を定義し，意味的に包含関係のもつ相・様相関数を新たな関数に置換する．そして，手順 (3) を行う．

手順 (3) 相・様相関数に任意記号を付ける．

本稿では，相・様相関数の個別の効果を調べるために，それぞれ汎化した関数ごとに新パターン辞書を作成する．また，これらの関数を同時に汎化した場合についても調べるため，使用頻度が 1,000 以上の相・様相関数に対し同時に汎化したパターン辞書も作成する．こうして，11 個のパターン辞書を作成する．

表 3: 時制・相・様相関数

表記		主な意味	日本語例	接続する要素[*1]	使用頻度
.kakodarou	*.kako*	過去	た, だ	*V*	35,392(39.88%)
	.darou	推量	だろう	*N*,*V*	54(0.06%)
.teiru		ている	*V* ている,*V* でいる	*V*	9,565(10.78%)
.dadantei	*.da*	断定（体言）	だ, である	*V*	7,163(8.07%)
	.dantei	断定（用言）	だ, である	*V*, *AJ*, *AJV*	3(0.00%)
.hitei		否定	ない, ぬ, くない	*V*, *AJ*, *AJV*	6,731(7.58%)
.rerurareru	*.reru*	可能, 受身, 自発, 尊敬	れる, られる	*V*	5,533(6.23%)
	.rareru	可能, 受身, 自発, 尊敬	れる, られる	*V*	1,845(2.08%)
.teinei		丁寧	ます	*N*, *V*, *AJ*, *AJV*	4,302(4.85%)
.you		意志	よう, う	*V*	2,734(3.80%)
.suiteirashii	*.suitei*	推定	ようだ	*V*	1,722(1.94%)
	.rashii	推定	らしい	*N*, *V*, *AJ*, *AJV*	57(0.06%)
.meireigotekudasai	*.meireigo*	命令語	なさい, たまえ	*V*	1,607(1.81%)
	.tekudasai	てください	*V* てください	*V*	1,197(1.35%)
.sase		使役	せる, させる	*V*	1,199(1.35%)
.tekuru		てくる	*V* てくる,*V* でくる	*V*	1,071(1.21%)
.tai		希望	たい	*V*	961(1.08%)
.tekureru		受益	*V* てくれる,*V* でくれる	*V*	954(1.07%)
.joutaihenka		状態変化	*AJ* くなる,*AJV* になる	*AJ*,*AJV*	891(1.00%)
.teshimatta		好ましくない結果＋過去	てしまった	*V*	724(0.82%)
.gimu		義務	べき, べきだ, べきである *V* ないといけない	*V*	574(0.65%)
.noda		のだ	のだ, のである	*V*, *AJ*, *AJV*	535(0.60%)
.temiru		てみる	*V* てみる,*V* でみる	*V*	419(0.47%)
.teiku		ていく	*V* ていく,*V* でいく	*V*	409(0.46%)
.teoku		ておく	*V* ておく,*V* でおく	*V*	391(0.44%)
.teshimau		好ましくない結果	てしまう	*V*	377(0.42%)
.sugiru		すぎる	用言, すぎる	*V*, *AJ*, *AJV*	326(0.37%)
.dekiru		可能	できる	*V*	314(0.35%)
.souda		不確実な断定, 伝聞	*V* そうだ	*V*, *AJ*, *AJV*	308(0.35%)
.utosuru		意志	うとする, ようとする	*V*	283(0.32%)
.teyaru		与益	*V* てやる,*V* でやる	*V*	212(0.24%)
.desu		丁寧な断定	です	*N*	206(0.23%)
.tearu		てある	*V* てある,*V* である	*V*	195(0.22%)
.kaisi		開始	*V* 始める,*V* 出す	*V*	179(0.20%)
.nisuru		使役+状態変化	*AJ* くする,*AJ* にする	*AJ*, *AJV*	173(0.19%)
.yotei		予定	*V* ことにする,*V* つもり	*V*, *AJ*, *AJV*	64(0.07%)
.teyoi		許可	てよい	*V*	49(0.06%)
.hougayoi		良好	ほうがよい, ほうがいい	*V*, *AJ*, *AJV*	36(0.04%)
.kirezu		破綻	*V* きれぬ,*V* きれず	*V*	1(0.00%)
合計					88,756(100.00%)

*1 別の様相関数に接続する場合もある

3.2.2　相・様相関数の汎化の例

以下に相・様相関数を汎化した時の様子を示す.

使用頻度が 1,000 以上の相・様相関数に対し同時に汎化

手順 (1) 相・様相関数の中で意味的に包含関係をもつ他の相・様相関数が存在するか調べる．存在する場合は，手順 (2) を行い，存在しない場合は，手順 (3) を行う.

$WJ234653$-01: /y $GEN1$! $NUM2$\$1^{ /tk $N3$ は }#4[/cf $ADV5$]/tk $N6$ について /cf $V7$^rentai !ために\$1 /cf($V8$ $\boxed{.reru.teiru.teinei}$| $ND8$ をされています)。

手順 (2) 相・様相関数の中で意味的に包含関係のもつ相・様相関数を統合し新たな関数を定義し，意味的に包含関係のもつ相・様相関数を新たな関数に置換する．そして，手順 (3) を行う.

$WJ234653$-01: /y $GEN1$! $NUM2$\$1^{ /tk $N3$ は }#4[/cf $ADV5$] /tk $N6$ について /cf $V7$^rentai !ために\$1 /cf($V8$ $\boxed{.rerurareru.teiru.teinei}$| $ND8$ をされています)。

手順 (3) 相・様相関数に任意記号を付ける.

$WJ234653$-01: /y $GEN1$! $NUM2$\$1^{ /tk $N3$ は }#1[/cf $ADV5$] /tk $N6$ について /cf $V7$^rentai !ために\$1 /cf($V8$ $\boxed{\#2[.rerurareru]\#3[.teiru]\#4[.teinei]}$| $ND8$ をされています)。

3.3　時制・相・様相の汎化の実施

3.3.1　時制・相・様相関数の汎化の方法

時制関数と使用頻度が 1,000 以上の相・様相関数に対し同時に汎化したパターン辞書も作成する.

時制関数と使用頻度が 1,000 以上の相・様相関数に対し以下の手順で汎化を行う.

手順 (1) 時制関数がパターンに記述されているに記述されているかを調べ，記述されている場合，手順 (2) を，記述されていない場合，手順 (3) を行う.

手順 (2) 時制関数を，自由時制関数に置換する．そして，手順 (5) を行う.

手順 (3) 基準パターン辞書から，時制関数と関数が組み合わさって記述されている述部の中で，時制関数が記述されている位置を調べる（表 2）．そして，[3] と表 1 より，パターンの述部に自由時制関数を挿入する位置を決める．そして，手順 (4) を行う．

手順 (4) 表 2 より，パターンの述部に自由時制関数を挿入する．そして，手順 (5) を行う．

手順 (5) 相・様相関数の中で意味的に包含関係をもつ他の相・様相関数が存在するか調べる．存在する場合は，手順 (6) を行い，存在しない場合は，手順 (7) を行う．

手順 (6) 相・様相関数の中で意味的に包含関係のもつ相・様相関数を統合し新たな関数を定義し，意味的に包含関係のもつ相・様相関数を新たな関数に置換する．そして，手順 (7) を行う．

手順 (7) 相・様相関数に任意記号を付ける．

3.3.2 時制・相・様相関数の汎化の例

以下に時制・相・様相関数を汎化した時の様子を示す．

時制関数と使用頻度が 1,000 以上の相・様相関数に対し同時に汎化

手順 (1) 時制関数がパターンに記述されているに記述されているかを調べ，記述されている場合，手順 (2) を，記述されていない場合，手順 (3) を行う．

$WJ234653\text{-}01$: /y $GEN1$! $NUM2\$1\hat{}\{$ /tk $N3$ は $\}\#4[$ /cf $ADV5]$ /tk $N6$ について /cf $V7\hat{}$rentai !ために$1 /cf($V8.reru.teiru.teinei|$ $ND8$ をされています)。

手順 (3) 基準パターン辞書から，時制関数と関数が組み合わさって記述されている述部の中で，時制関数が記述されている位置を調べる（表 2）．そして，[3] と表 1 より，パターンの述部に自由時制関数を挿入する位置を決める．そして，手順 (4) を行う．

$WJ234653\text{-}01$: /y $GEN1$! $NUM2\$1\hat{}\{$ /tk $N3$ は $\}\#4[$ /cf $ADV5]$/tk $N6$ について /cf $V7\square\hat{}$rentai !ために$1 /cf($V8.reru.teiru.teinei\square|$ $ND8$ をされています)。

手順 (4) 表 2 より，パターンの述部に自由時制関数を挿入する．そして，手順 (5) を行う．

$WJ234653\text{-}01$: /y $GEN1$! $NUM2\$1\hat{}\{$ /tk $N3$ は $\}\#4[$ /cf $ADV5]$ /tk $N6$ について /cf $V7\boxed{\#1[.kakodarou]}\hat{}$rentai !ために$1 /cf($V8.reru.teiru.teinei\boxed{\#1[.kakodarou]}|$ $ND8$ をされています)。

手順 (5) 相・様相関数の中で意味的に包含関係をもつ他の相・様相関数が存在するか調べる．存在する場合は，手順 (6) を行い，存在しない場合は，手順 (7) を行う．

$WJ234653\text{-}01$: /y $GEN1$! $NUM2\$1\hat{}\{$ /tk $N3$ は $\}\#4[$ /cf $ADV5]$ /tk $N6$ について /cf $V7\#1[.kakodarou]\hat{}$rentai !ために$1 /cf($V8\boxed{.reru.teiru.teinei}\#1[.kakodarou]|$ $ND8$ をされています)。

手順 (6) 相・様相関数の中で意味的に包含関係のもつ相・様相関数を統合し新たな関数を定義し，意味的に包含関係のもつ相・様相関数を新たな関数に置換する．そして，手順 (7) を行う．

$WJ234653\text{-}01$: /y $GEN1$! $NUM2\$1\hat{}\{$ /tk $N3$ は $\}\#4[$ /cf $ADV5]$ /tk $N6$ について /cf $V7\#1[.kakodarou]\hat{}$rentai !ために$1 /cf($V8\boxed{.rerurareru.teiru.teinei}\#1[.kakodarou]|$ $ND8$ をされています)。

手順 (7) 相・様相関数に任意記号を付ける．

$WJ234653\text{-}01$: /y $GEN1$! $NUM2\$1\hat{}\{$ /tk $N3$ は $\}\#4[$ /cf $ADV5]$ /tk $N6$ について /cf $V7\#1[.kakodarou]\hat{}$rentai !ために$1 /cf($V8\boxed{\#1[.rerurareru]\#1[.teiru]\#1[.teinei]}\#1[.kakodarou]|$ $ND8$ をされています)。

3.4 位置変更可能記号の無効化の実施

3.4.1 位置変更可能記号の無効化の方法

位置変更可能記号とは出現する位置が変わっても文型パターンとしての意味が変わらない表現要素について，出現できる位置を指定する記号 ($n\hat{}\{\cdots\}$,$n で表記) である．

基準パターン辞書には既に位置変更可能記号がパターンに挿入されている．位置変更可能記号に対し以下の手順で無効化を行う．

手順 (1) 文型パターンに位置変更可能記号が記述されている場合，手順 (2) を行う．

手順 (2) 文型パターンに記述されている位置変更可能記号を取り除く．

本稿では，基準パターン辞書から位置変更可能記号を取り除いたパターン辞書を作成し，性能の降下を調べる．

3.4.2　位置変更可能記号の無効化の例

以下に位置変更可能記号の無効化した時の様子を示す.

位置変更可能記号の無効化

手順 (1) 文型パターンに位置変更可能記号が記述されている場合，手順 (2) を行う.

$WJ237906\text{-}01$: /y $\boxed{\$1\hat{}\{}$ /tk $N1$ は $\boxed{\}}$ /tcfk $N2$ が /tcfk $N3.da\hat{}$rentai ので $\boxed{\$1}$ ！ (始終 | しじゅう)$1 /f 気づかっている。

手順 (2) 文型パターンに記述されている位置変更可能記号を取り除く.

$WJ237906\text{-}01$: /y /tk $N1$ は /tcfk $N2$ が /tcfk $N3.da\hat{}$rentai ので！ (始終 | しじゅう) /f 気づかっている。

3.5　汎化・無効化の結果

基準パターン辞書のパターン数と，項目 3.1，項目 3.2，項目 3.3，項目 3.4 のパターン数の規模を表 4 にまとめる.

表 4: パターン辞書の規模

項目 / パターン辞書	パターン辞書のパターン数の規模	パターン辞書のバイト数
基準パターン	122,619	12,814,141
位置変更可能記号を無効化	84,747	12,523,409
時制関数を汎化	324,878	14,871,877
$[.teiru]$	132,457	12,869,241
$[.rerurareru]$	130,366	12,900,656
$[.dadantei]$	129,898	12,888,466
$[.hitei]$	129,513	12,850,778
$[.teinei]$	127,147	12,834,840
$[.meireigotekudasai]$	125,510	12,851,035
$[.you]$	125,445	12,826,589
$[.suiteirashii]$	124,426	12,836,158
$[.sase]$	123,855	12,824,395
$[.tekuru]$	123,708	12,819,909
出現頻度が 1,000 以上の相・様相関数を同時に汎化	181,781	13,196,238
時制関数と出現頻度が 1,000 以上の相・様相関数を同時に汎化	582,680	15,591,503

4　汎化・無効化の効果の実験

4.1　実験の目的

文型パターン辞書は汎化前のパターン辞書（基準パターン辞書）と，第 3 章で作成した汎化後のパターン辞書（対象パターン辞書）を使用し，汎化の効果を第 2 章の評価パラメータを用いて定量的に評価する．

4.2　実験の方法

実験には，実際に文を入力して，パターンの被覆率を求める．テスト用の入力文は形態素解析結果 123,016 文 ($=I$) を使用する．入力文と文型パターン辞書を文型パターンパーサ [5] を用いて照合を行い，照合結果から，以下の項目 (i)〜(vi) を調べる．

(i) アンマッチ：文法・単語レベル 123,016 文中どのパターンにもマッチしなかった文数

(ii) 自己パターンにのみマッチ：テスト文から作成された文型パターンにのみマッチした文数

(iii) 他パターンにのみマッチ：テスト文から作成された文型パターン以外の文型パターンにのみマッチした文数

(iv) 自己パターン他パターン共にマッチ：テスト文から作成された文型パターンとテスト文から作成された文型パターン以外の文型パターンが共にマッチした文数

(v) 完全一致適合パターン数 P_{N1}：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の総数

(vi) 部分一致適合パターン数 P_{N2}：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の総数

「自己文型パターン」以外の適合文型パターンが 1 つ以上存在した入力文の数 M は (ii) と (iv) を足したもので，適合文型パターン数 P は (v) と (vi) を足したものである．文型再現率 $R1$ と平均適合パターン数 N の実測値は，それぞれ (1) 式，(5) 式にテスト用入力文の数 I，M，P の実測値を代入して求める．

[2] より，基準パターン辞書の被覆率のパラメータ $R1$ および N の近似式の係数は，以下のようになる．

$$\lambda_1 = 0.005038 \tag{9}$$

$$\lambda_2 = 0.4171 \tag{10}$$

$$\lambda_3 = 13.4085/122,619.0 \tag{11}$$

したがって，$R1$ および N の実測値と近似式の係数を (7) 式，(8) 式に代入し，p_{R1} および p_N を求め，p_{R1} および p_N を X とし，(6) 式より η_{R1} と η_N を求める．

4.3 実験の様子

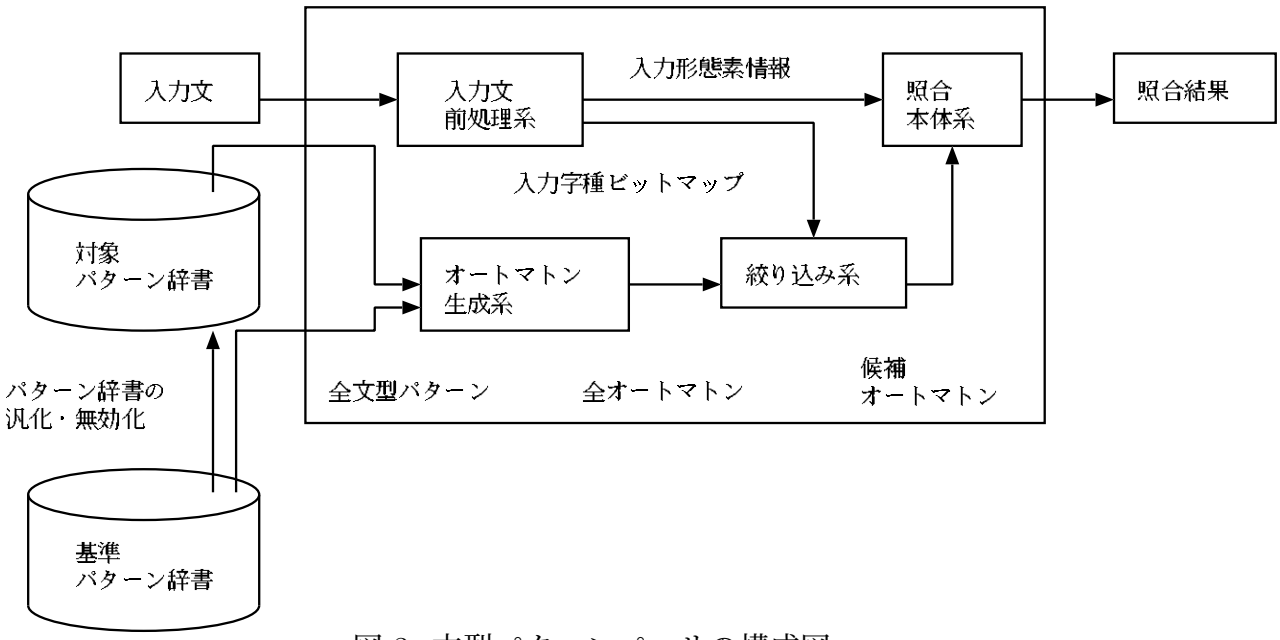

図 3: 文型パターンパーサの構成図

文型パターンパーサ [5] を使用した時の入力文と照合結果の例を示す.

• 入力文を以下に示す.

「海上保険業者は委付を受諾することに決定した。」

照合結果の出力ファイルの形式の仕様を以下に示す.

1. INPUT=「入力形態素ファイル名」=「文字位置ビット」=「形態素位置ビット」
2. 入力文
3. 形態素情報
4. 入力情報の区切り記号
5. PATTERN=「適合パターン番号」=「パターン記述」=「経路情報」=「適合文字位置ビット」=「適合形態素位置ビット」
6. 変数名=「標準形表記」=「原文表記」=「経路情報」=「適合文字位置ビット」=「適合形態素位置ビット」

● 照合結果の出力ファイルを以下に示す.

INPUT=BA000002-00=0x3fffff=$\underline{0x7ff}_{(3)}$
海上保険業者は委付を受諾することに決定した。
1. /海上保険 (1100,{NI:1170,NK:88})
2. +業者 (1100,{NI:273,NI:268,NI:374,KR:4308s05})
3. +は (7530)
4. /委付 (1220,NI:1725)
5. +を (7430)
6. /受諾する (2437,NI:1732,NY:32,KR:4208a65)
7. /こと (1800, 事,NI:1235,KR:9904h00)
8. +に (7430)
9. /決定し (2433, 決定する,NI:1442,NI:1022,NY:20,NY:32,NY:5,KR:1700a02)
10. +た (7216)
11. +。([P]0110)
12. /nil

PATTERN=$\underline{WJ161968}_{(1)}$-01= /y </tk $N1$ は> /cf $N2$ を /cf($V3$^rentai|$V3$.*kakodarou*^rentai) ! ことに /cf ($V4$|$V4$.*kakodarou*|$V4$.*teinei*|$V4$.*teinei.kakodarou*|$ND4$ をしました)。
=[/y , /tk ,$N1$, は, /cf,$N2$, を, /cf ,$V3$, ! , ことに,/cf ,$V4$,.*kakodarou*,。]
=$\underline{0x3fffff}_{(2)}$=$\underline{0x7ff}_{(4)}$
/y===?=0x000000=0x000
/tk===?=0x000000=0x000
N1=海上保険業者=海上保険業者=n(ncx(1),ncxmk(1),ncx(2))=0x00003f=0x003
/cf===?=0x000000=0x000
N2=委付=委付=n(ncx(4))=0x000180=0x008
/cf===?=0x000000=0x000
V3=受諾する=受諾する=v(ve(6))=0x003c00=0x020
/cf===?=0x000000=0x000
V4=決定する=決定し=v(ve(9))=0x0e0000=0x100

PATTERN=$\underline{WJ188987}_{(1)}$-01=/y</tk$N1$は>#1[/cf$GEN3$]/k$N4$は/cf ($V5$^rentai|$V5$.*kakodarou*^rentai|$ND5$ をする) ! ことに /cf $V6$#2[.*kakodarou*]。=[/y , /k ,$N4$, は, /cf ,$V5$,! , ことに, /cf ,$V6$,.*kakodarou*,。]
=$\underline{0x3ffc7f}_{(2)}$=$\underline{0x7e7}_{(4)}$
/y===?=0x000000=0x000
/tk===?=0x000000=0x000
N4=海上保険業者=海上保険業者=n(ncx(1),ncxmk(1),ncx(2))=0x00003f=0x003
V5=受諾する=受諾する=v(ve(6))=0x003c00=0x020
/cf===?=0x000000=0x000
V6=決定する=決定し=v(ve(9))=0x0e0000=0x100
#2=[.kakodarou]=0x100000=0x200

照合結果の出力ファイルから評価パラメータを求めるには，まず照合結果の適合パターンの中から，自己パターンと他パターンを分ける．自己文型パターンと他文型パターンを分けるには，入力形態素ファイル名と入力文の自己パターンのパターン番号がある DB を用い，照合結果のアンダーライン (1) と比べ，適合パターン番号が同じなら自己パターン，適合パターン番号が異なるなら他パターンとなる．そして，以下の方法で照合結果から評価パラメータの式の値を求める．

•$R1$ は，照合結果に他パターンあれば M の値が 1 つ増える．

•$R2$ は，照合結果の中で，自己パターン以外のアンダーライン (2) の値（16 進数）を調べ，その中で最も高い値を Mj とする．

• アンダーライン (3) とアンダーライン (4) を比べ，値が同じなら P_{N1} が 1 つ，値が異なるのなら P_{N2} が 1 つ増える．

このように，入力文毎の照合結果から求めた値を (1)〜(4) 式に代入し，それぞれの評価パラメータを求める．

4.4 実験の結果

実験の結果を表 5，表 6，表 7 にまとめる（η_d については考察で述べる）．表 7 より，汎化の効果が最も高いものは辞書 (14) の時制・相・様相を同時に汎化したパターン辞書だとわかった．

表 5: 照合の結果

測定項目 / パターン辞書	アンマッチ (文数/%)	自己パターンにのみマッチ (文数/%)	他パターンにのみマッチ (文数/%)	自己パターン他パターン共にマッチ (文数/%)
基準パターン	1,217/0.99	62,461/50.77	280/0.23	59,058/48.01
(1) 位置変更可能記号を無効化	1,220/0.99	62,969/51.19	272/0.22	58,555/47.60
(2) 時制関数を汎化	1,219/0.99	54,897/44.63	335/0.27	66,565/54.11
(3)[*.teiru*]	1,195/0.97	59,499/48.37	302/0.25	62,020/50.42
(4)[*.rerurareru*]	1,215/0.99	61,418/49.93	283/0.23	60,100/48.86
(5)[*.dadantei*]	1,213/0.99	62,132/50.51	285/0.23	59,386/48.28
(6)[*.hitei*]	1,215/0.99	61,727/50.18	282/0.23	59,792/48.61
(7)[*.teinei*]	1,214/0.99	61,934/50.35	283/0.23	59,585/48.44
(8)[*.meireigotekudasai*]	1,207/0.98	61,702/50.16	289/0.23	59,818/48.63
(9)[*.you*]	1,214/0.99	62,106/50.49	283/0.23	59,413/48.30
(10)[*.suiteirashii*]	1,204/0.98	61,639/50.11	293/0.24	59,880/48.68
(11)[*.sase*]	1,217/0.99	62,388/50.72	280/0.23	59,131/48.07
(12)[*.tekuru*]	1,217/0.99	62,397/50.72	280/0.23	59,122/48.06
(13) 出現頻度が 1,000 以上の相・様相関数を同時に汎化	1,193/0.97	55,462/45.09	347/0.28	66,014/53.66
(14) 時制関数と出現頻度が 1,000 以上の相・様相関数を同時に汎化	2,861/2.33	46,924/38.14	1,247/1.01	71,984/58.52

表 6: 評価パラメータの結果 (1)

測定項目 / パターン辞書	M^{*2}	適合文型パターン数 P	文型再現率 $R1$	文型一致率 $R2$	$N1^{*3}$	$N2^{*4}$	平均適合パターン数 N
基準パターン	59,338	1,649,455	48.24	31.44	1.89	11.53	13.41
(1) 位置変更可能記号を無効化	58,827	1,570,839	47.82	31.03	1.79	10.98	12.77
(2) 時制関数を汎化	66,900	2,244,275	54.38	35.94	2.40	15.85	18.24
(3)[*.teiru*]	62,322	1,818,128	50.66	32.58	1.97	12.80	14.78
(4)[*.rerurareru*]	60,383	1,775,492	49.09	32.13	2.00	12.43	14.43
(5)[*.dadantei*]	59,671	1,676,581	48.51	31.68	1.89	11.74	13.63
(6)[*.hitei*]	60,074	1,679,508	48.83	31.92	1.90	11.75	13.65
(7)[*.teinei*]	59,868	1,697,710	48.67	31.79	1.91	11.89	13.80
(8)[*.meireigotekudasai*]	60,107	1,674,613	48.46	31.82	1.90	11.72	13.61
(9)[*.you*]	59,696	1,678,091	48.53	31.67	1.90	11.74	13.64
(10)[*.suiteirashii*]	60,173	1,659,082	48.91	31.77	1.88	11.60	13.49
(11)[*.sase*]	59,411	1,728,212	48.30	31.52	2.00	12.05	14.05
(12)[*.tekuru*]	59,402	1,677,447	48.29	31.53	1.90	11.74	13.64
(13) 出現頻度が 1,000 以上の相・様相関数を同時に汎化	66,361	2,304,239	53.95	35.25	2.41	16.32	18.73
(14) 時制関数と出現頻度が 1,000 以上の相・様相関数を同時に汎化	73,231	3,774,328	59.53	39.65	3.52	27.16	30.68

表 7: 評価パラメータの結果 (2)

測定項目 / パターン辞書	$R1$ からみた η η_{R1}	N からみた η η_N	η_d^{*5}
基準パターン	1.00	1.00	1.00
(1) 位置変更可能記号を無効化	0.98	0.95	0.69
(2) 時制関数を汎化	1.53	1.36	2.65
(3)[*.teiru*]	1.19	1.10	1.08
(4)[*.rerurareru*]	1.06	1.08	1.06
(5)[*.dadantei*]	1.02	1.02	1.06
(6)[*.hitei*]	1.05	1.02	1.06
(7)[*.teinei*]	1.03	1.03	1.04
(8)[*.meireigotekudasai*]	1.05	1.02	1.02
(9)[*.you*]	1.02	1.02	1.02
(10)[*.suiteirashii*]	1.07	1.01	1.01
(11)[*.sase*]	1.01	1.05	1.01
(12)[*.tekuru*]	1.01	1.02	1.01
(13) 出現頻度が 1,000 以上の相・様相関数を同時に汎化	1.49	1.40	1.48
(14) 時制関数と出現頻度が 1,000 以上の相・様相関数を同時に汎化	2.15	2.29	4.75

*2「自己文型パターン」以外の適合文型パターンが 1 つ以上存在した入力文の数
*3 完全一致平均適合パターン数
*4 部分一致平均適合パターン数
*5 パターンの記述量の違いからみた η

5　考察

5.1　η の積の関係

辞書 (3) から辞書 (12) を同時に汎化したものが辞書 (13) である．辞書 (2) と辞書 (13) を同時に汎化したものが辞書 (14) である．η_{R1}，η_N はそれぞれ辞書 (3) から辞書 (12) の η の積をとると辞書 (13) の η に近い．辞書 (2) と辞書 (13) の η の積をとると辞書 (14) の η に近い．これは，汎化された部分がそれぞれのパターン辞書において独立しているためだと考えられる．

これより，それぞれの時制・相・様相関数を個別に汎化し，それぞれの汎化の効果の積をとれば，時制・相・様相関数を同時に汎化したパターン辞書の汎化の効果が予測できると考えられる．

5.2　パターン辞書を汎化しない推定法

η_{R1} と η_N のように近似的に対象辞書を構築せず，汎化の効果を予想する推定法を検討する．そこで，本稿では,「時制・相・様相」のパターン辞書上の使用頻度に関して求めた η_d を提案する．

η_d を求めるには，1 パターンに対する汎化した箇所を求める．例えば，1 パターンに対し 1 箇所汎化をした時，そのパターンは 2 倍に増えたことになる．また，2 箇所汎化をした時は，そのパターンは 4 倍に増えたことになる．

以下に 2 箇所汎化した時の具体例を示す．

（汎化前） $N1$ が $V2.teiru.hitei$。

（汎化後） $N1$ が $V2$♯1[.*teiru*]♯2[.*hitei*]。

（汎化後のパターンの展開 1） $N1$ が $V2$。

（汎化後のパターンの展開 2） $N1$ が $V2.teiru$。

（汎化後のパターンの展開 3） $N1$ が $V2.hitei$。

（汎化後のパターンの展開 4） $N1$ が $V2.teiru.hitei$。

このように，汎化したことによって増えたパターンの総数を求め，その総数を基準パターンのパターン数で割ったものを η_d とする．汎化したことによって増えたパターンの総数は，すでに表 4 にまとめている．それぞれのパターン辞書の η_d の値を表 7 にまとめる．

表 7 より，辞書 (3) から辞書 (13) の η_{R1}，η_N の値は η_d の値と近似している．しかし，辞書 (2) と辞書 (14) の η_{R1}，η_N は η_d と値がかなり異なっている．これは，可能な限り時制を汎化したので，入力文と照合しないパターンが多数存在したためだと考えられる．

これより η_d は相・様相の汎化では汎化の効果の予測ができる．しかし，時制の汎化では汎化の効果の予測はできないと考えられる．

5.3 規模の異なるパターン辞書の汎化による推定法

大規模な被覆率調査を行うと大量の作業コストがかかる．そこで，規模の異なる被覆率調査行い，あらかじめ汎化の効果を予測する推定法を検討する．大規模な被覆率調査で求めた η と，規模の異なる被覆率調査で求めた η を比較し，規模の異なる被覆率調査でも，大規模な被覆率調査を行ったときと同等の汎化の効果を予測できるか確認する．

本稿では，規模の異なる自由時制パターン辞書を対象パターン辞書とし，規模の異なるパターン辞書ごとに η を算出し，自由時制パターン辞書の全パターン数の η と比較を行う．それぞれの η の値を表 8 にまとめる．

表 8: 規模の異なる自由時制パターン辞書の η

小規模パターン数	規模の異なるパターン数の		
	η_{R1}	η_N	η_d
100	0.01	0.15	2.72
500	0.13	0.15	2.82
1,000	4.07	1.12	2.64
5,000	1.64	1.30	2.63
10,000	1.83	1.40	2.70
20,000	2.05	1.34	2.67
40,000	1.72	1.47	2.69
80,000	1.69	1.39	2.67
全パターン数	全パターン数の		
	η_{R1}	η_N	η_d
122,619	1.53	1.36	2.65

その結果，80,000 パターンの η が全パターン数の η に最も近い値となった．ただし，小数点第 1 位の誤差を許せば，40,000 パターンでも汎化の効果が予測できると考えられる．100 から 20,000 パターンでは，データが大幅に変動するので，値に信頼性が持てない．η_d

はパターン数が変動しても値が安定しているが，5.2 で述べたように，時制の汎化に関しては汎化の効果は予測ができない．

これより，規模の異なる被覆率調査で求めた実測値 η も汎化の効果の予測ができる可能性があることがわかった．今後様々に考えられる汎化の方法の中で，効果の見込まれる汎化の方法の選別に有効である．

6　おわりに

本稿では，研究の背景として，パターン辞書の構築の際，パターンの汎用性を向上するため，記述要素の効果的な汎化が求められていたことに対し，本稿では，以下の記述要素の汎化を実行した．

「要素の位置変更可能の指定」,「時制・相・様相を表す関数の汎化」をする場合としない場合についての被覆率調査を行い，η を用いて定量的に評価した．実験結果より,「要素の位置変更可能の指定」,「時制・相・様相を表す関数の汎化」をすることは，被覆率が 20%上昇し，文型パターン数が 2 倍に拡大するので，パターンの汎用性を向上するために有効だとわかった．また，汎化の効果があらかじめ予想できることを確認した．

今後は，各 η の異なりが生じる原因を定性的に考察し，単語レベルだけではなく，句レベル，節レベルの文型パターンの汎化を行い，汎化の効果を調査する予定である．

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深く感謝します.

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深く感謝の意を表します.

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] 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,2005(発表予定).

[3] 南不二男:現代日本語文法の輪郭, 大修館書店,1993.

[4] 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.

[5] 徳久雅人, 池原悟, 村上仁一:文型パターンパーサの試作, 言語処理学会第 10 回年次大会発表論文集,pp.608-611,2004.