概要

近年，日英機械翻訳方式として，パターン翻訳方式が注目されている．そして，現在，24 万件の文型パターン辞書が構築されている．ところが，規模が大きい為，パターン数を削減する方法が必要とされているが，人手によりパターン数を削減する事は困難である．

本研究では，パターン間の包含関係に着目して，文型パターン辞書のパターン数を自動的に削減する方法を提案した．また，実際にパターン数の削減を行った．提案した方法により，パターン間の包含関係を判定した結果，包含関係を持つパターンは 12,981 パターン (10.6%) あった．そのうち，削除したパターンは 9,852 パターン (8.0%) であった．しかし，削除が行えないパターンの多くが，適合頻度（適合した原文の数）が少ないものであった．

今後，パターン数の削減を試みる場合は，包含関係だけでなく類似度の高いパターンにも着目し，新たなパターンを作成するなどしてパターン数の削減を試みる方法があると考えられる．また，包含関係を持たないパターンについて削減の必要性を検討する必要があることも分かった．

目次

表目次

1　はじめに

近年，日英機械翻訳の方式の１つとして，パターン翻訳方式が注目されている．従来までは翻訳に用いられるパターン数が少なかった為に，特定の狭い分野の翻訳に使われることが多かった．しかし，参考文献 [1] で提案された方法により，24 万件の大規模文型パターン辞書が構築されている．この文型パターン辞書は，15 万件の日英対訳コーパスを元に作成されている．ところが，この文型パターン辞書は規模が大きい為，実際に翻訳を行う場合に検索コストかかる等の問題がある．そのため，パターン数の削減が必要とされているが，人手によるパターン数の削減は規模を困難である．

そこで，本研究では日本語文型パターン間の包含関係に着目して，その数を半自動的に削減する方法を提案する．また，その方法を用いて文型パターン辞書の縮退を試みる．

提案した方法により，パターン間の包含関係を判定した結果，包含関係を持つパターンは 12,981 パターン (10.6%) あった．そのうち，削除したパターンは 9,852 パターン (8.0%) であった．

本論文の構成は以下の通りである．第 2 章では，文型パターン辞書と包含関係について説明する．第 3 章では，削減方法を順を追って説明する．第 4 章では，包含関係による削減の実験について説明する．第 5 章では，包含関係のよる削減の実験結果と具体例を示す．第 6 章では，削減結果についての考察を説明する．第 7 章では結論と今後の課題を述べる．

2　文型パターン辞書と包含関係

本研究で用いる文型パターン辞書とその記述言語，包含関係について次に示す．

2.1　文型パターン辞書

文型パターン辞書は，15 万件の日英対訳コーパスを元に，非線形要素を半自動的に変数・関数化することで作成されている．

2.2　パターン記述言語

文型パターンを構成する要素には，字面，変数，関数，記号がある．[2]

変数には，名詞を表す N，動詞を表す V，名詞句を表す NP などがある．(表 1)

関数は，変数に適合する値の形式や，字面の指定，表現の統括を行う．(表 2)

記号は，パターン要素の適合の仕方について，任意化，選択，順序変更，記憶という制御を行う．(表 3, 表 4)

表 1 変数一覧

変数名	意　味	変数名	意　味
単語レベル		**句レベル**	
N	名詞	*NP*	名詞句
ND	「する」に先行する名詞	*VP*	動詞句
NUM	数詞	*AJP*	形容詞句
TIME	時詞	*AJVP*	形容動詞句
V	動詞	*ADVP*	副詞句
AJ	形容詞	**節レベル**	
AJV	形容動詞	*CL*	節
ADV	副詞		
REN	連体詞		
GEN	限定詞		

表 2 関数一覧

種類	意味と例
様相関数	変数に後続する表現の形式を指定
	例) *V*1.*reru*
語尾関数	変数に対応する表現の形式を指定
	例) *V*1^*katei*
字面関数	変数に含まれる字面を指定
	例) #大変 (*CL*1)
マクロ関数	引数内表現を指定する変数が統括
	例) #*CL*1(*N*2 が *V*3)

表 3 記号一覧

記号名	表記	意味
離散記号	/...	文型に無関係な要素 (適合条件については表**??**参照)
選択記号	(…)	いずれかの要素列と適合
任意記号	[…]	文型選択上，任意の要素
補完要素記号	<…>	ゼロ代名詞等
順序任意要素指定記号	{…,…}	順序入れ替え可能な範囲
位置変更可能要素指定記号	$n^{…}, $n	指定位置に入れ替え可能
文節境界記号	!	文節の境界と適合
記憶記号	#n	適合内容を記憶

表 4 離散記号の適合条件

表記	適合条件
y	連用節
t	連体節
c	格要素
f	連用修飾句（副詞句）
k	連体修飾句（形容詞，形容動詞連体形，連体詞）

2.3 パターン間の包含関係

パターン $P1$ に適合する入力文の全てがパターン $P0$ に適合する時，パターン $P1$ はパターン $P0$ に包含されると定義し，また，パターン $P0$ を上位のパターン，パターン $P1$ を下位のパターンと呼ぶ．両者の関係を $P0 \supseteq P1$ と表記する．

以下に包含関係にあるパターンの例を示す．

$P0$: $N1$ は [$N2$ を]$VP3$。
$P1$: 私は $V1$。

2.4 パターン要素間の包含関係

パターン間の包含関係を判定するためには，全ての入力文に対して適合の可否を調査する必要がある．しかし，全ての入力文に対して調査は不可能である．

そこで，パターン自身が適合可能な入力文の領域を表している事に着目して，パターンが別のパターンに適合するかどうかを調査することで，包含関係を判定する．

その為には，パターンを構成する要素（変数，関数，記号，字面）間の包含関係を定義する必要がある．そこで，各要素の定義に基づいて包含関係を定義する．表 5 に要素間の関係の一部を示す．また，全ての定義を付録に示す．

表 5 パターン要素間の包含関係 (一部)

上位の変数・記号	下位
N	*NUM*, *TIME*, *ND*
NP	*N*, *N* の下位要素
VP	*V*
AJP	*AJ*
AJVP	*AJV*
ADVP	*ADV*
/	すべての離散記号
/cf	/c, /f
/tk	/t, /k
/yf	/y, /f
/tck	/t, /c, /k
/ytk	/y, /t, /k, /tk
/tcfk	/t, /c, /k, /f, /cf, /tk, /tck

2.5 パターンパーサ

文型パターンパーサ (以下パーサ)[3] は，日本語文型パターンと日本語文との照合を行うプログラムである．照合方式は，ATN(Augmented Transition Network)[4] をベースとしている．

本研究では，次に示す理由により既存のパーサを用いて包含関係の判定を行う．

- 既に定義されているパターン要素のオートマトン定義が利用出来る為，日本語単語と変数との間の照合結果が保証されている
- オートマトン定義を変更する事で容易に変数・関数が適合する対象の変更が可能
- パターン解析などのプログラム作成時間を短縮が可能

3　削減方法

本章では，提案するパターン数削減の手順を示す．

3.1　オートマトン定義の追加・修正

パーサには，パターン中の変数・関数が日本語形態素を受理する為のオートマトン定義が用意されている．しかし，本研究では入力文としての変数・関数をパターン側の変数・関数が受理出来る必要がある．そこで，表 5 で示した包含関係を元に，パーサのオートマトン定義を追加・修正する．句・節レベルの変数 (*NP*,*CL* など) では，単語レベルの変数 (*N*,*V* など) を用いてオートマトンの定義をしてある場合に，変数を含む句・節も受理可能になる．そうでない場合は，個別に定義の追加・修正が必要になる．

本研究では，98 個の変数・関数に対して 109 個の追加・修正を行った．修正後のオートマトン定義の例を図示したものを図 1 に示す．図中の IMPADVP が変数 *ADVP* を受理する為に追加された定義である．

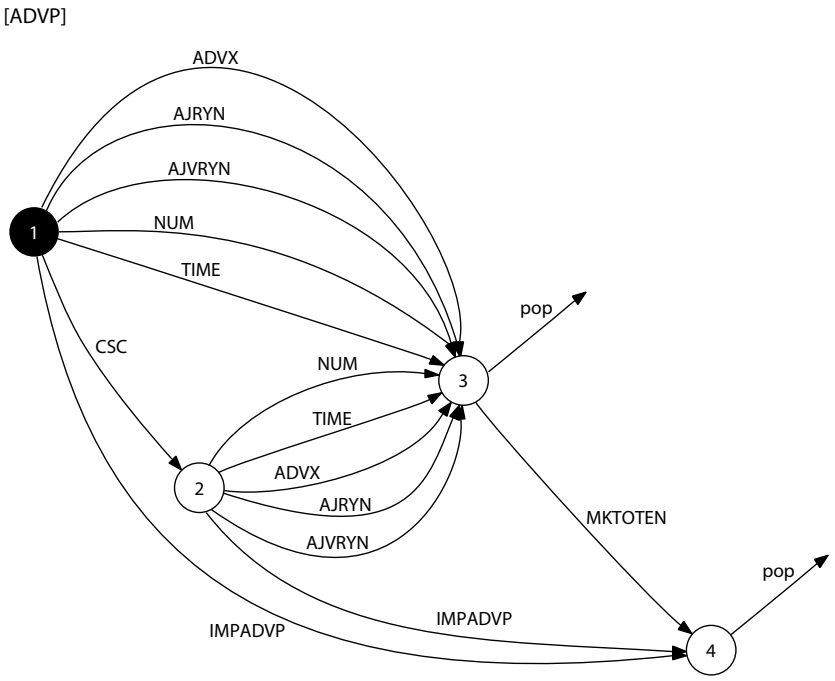

図 1 修正後のオートマトンの例 (副詞句変数 *ADVP*)

3.2 パターンの形態素解析

パーサは，日本語文の形態素解析データと日本語文型パターンを入力データとして受け付ける．しかし，本研究では日本語文型パターン間で照合を行う．そのため，パターンを日本語文の形態素解析データに見せかける必要がある．

そこで，パターンの形態素解析を行う為に，記号の展開と展開後パターンに対する形態素解析データの作成を行う．

3.2.1 記号の展開

日本語文には，パターンで用いられる，要素選択記号，任意記号等に対応する表現方法は存在しない．そこで，記号の定義を元にパターンの展開を行う．次に展開例を示す．

■展開例１
パターン：

/y < /tk *N*1 は > /tcfk *N*2 を /cf *V*3 (て | で) /cf (*V*4.*kako* | *ND*4 をした)。

展開後：

1. /y /tcfk *N*2 を /cf *V*3 て /cf *V*4.*kako*。
2. /y /tk *N*1 は /tcfk *N*2 を /cf *V*3 て /cf *V*4.*kako*。
3. /y /tcfk *N*2 を /cf *V*3 で /cf *V*4.*kako*。
4. /y /tk *N*1 は /tcfk *N*2 を /cf *V*3 で /cf *V*4.*kako*。
5. /y /tcfk *N*2 を /cf *V*3 て /cf *ND*4 をした。
6. /y /tk *N*1 は /tcfk *N*2 を /cf *V*3 て /cf *ND*4 をした。
7. /y /tcfk *N*2 を /cf *V*3 で /cf *ND*4 をした。
8. /y /tk *N*1 は /tcfk *N*2 を /cf *V*3 で /cf *ND*4 をした。

■展開例２
パターン：

/y $1^{/tk *N*1 は } /cf *V*2 ながら$1 /f (*V*3|*ND*3 をする)。

展開後：

1. /y /tk *N*1 は /cf *V*2 ながら /f *V*3。
2. /y /cf *V*2 ながら /tk *N*1 は /f *V*3。
3. /y /tk *N*1 は /cf *V*2 ながら /f *ND*3 をする。
4. /y /cf *V*2 ながら /tk *N*1 は /f *ND*3 をする。

■展開例３

パターン：

/y $1^{ /tk *N*1 は }#2[/cf (しっかり | 確り)]$1 /tk *N*3 を$1 /cf *V*4(て | で)$1 /tk *N*5 を $1 /cf *V*6.*kako*。

展開後：

1. /y /tk *N*1 は /tk *N*3 を /cf *V*4 て /tk *N*5 を /cf *V*6.*kako*。
2. /y /tk *N*1 は /cf しっかり /tk *N*3 を /cf *V*4 て /tk *N*5 を /cf *V*6.*kako*。
3. /y /tk *N*1 は /cf 確り /tk *N*3 を /cf *V*4 て /tk *N*5 を /cf *V*6.*kako*。
4. /y /cf しっかり /tk *N*1 は /tk *N*3 を /cf *V*4 て /tk *N*5 を /cf *V*6.*kako*。
5. /y /cf 確り /tk *N*1 は /tk *N*3 を /cf *V*4 て /tk *N*5 を /cf *V*6.*kako*。
6. /y /tk *N*3 を /tk *N*1 は /cf *V*4 て /tk *N*5 を /cf *V*6.*kako*。
7. /y /cf しっかり /tk *N*3 を /tk *N*1 は /cf *V*4 て /tk *N*5 を /cf *V*6.*kako*。
8. /y /cf 確り /tk *N*3 を /tk *N*1 は /cf *V*4 て /tk *N*5 を /cf *V*6.*kako*。
9. /y /tk *N*1 は /tk *N*3 を /cf *V*4 で /tk *N*5 を /cf *V*6.*kako*。
10. /y /tk *N*1 は /cf しっかり /tk *N*3 を /cf *V*4 で /tk *N*5 を /cf *V*6.*kako*。
11. /y /tk *N*1 は /cf 確り /tk *N*3 を /cf *V*4 で /tk *N*5 を /cf *V*6.*kako*。
12. /y /cf しっかり /tk *N*1 は /tk *N*3 を /cf *V*4 で /tk *N*5 を /cf *V*6.*kako*。
13. /y /cf 確り /tk *N*1 は /tk *N*3 を /cf *V*4 で /tk *N*5 を /cf *V*6.*kako*。
14. /y /tk *N*3 を /tk *N*1 は /cf *V*4 で /tk *N*5 を /cf *V*6.*kako*。
15. /y /cf しっかり /tk *N*3 を /tk *N*1 は /cf *V*4 で /tk *N*5 を /cf *V*6.*kako*。
16. /y /cf 確り /tk *N*3 を /tk *N*1 は /cf *V*4 で /tk *N*5 を /cf *V*6.*kako*。
17. /y /tk *N*3 を /cf *V*4 て /tk *N*1 は /tk *N*5 を /cf *V*6.*kako*。
18. /y /cf しっかり /tk *N*3 を /cf *V*4 て /tk *N*1 は /tk *N*5 を /cf *V*6.*kako*。
19. /y /cf 確り /tk *N*3 を /cf *V*4 て /tk *N*1 は /tk *N*5 を /cf *V*6.*kako*。
20. /y /tk *N*3 を /cf *V*4 で /tk *N*1 は /tk *N*5 を /cf *V*6.*kako*。
21. /y /cf しっかり /tk *N*3 を /cf *V*4 で /tk *N*1 は /tk *N*5 を /cf *V*6.*kako*。
22. /y /cf 確り /tk *N*3 を /cf *V*4 で /tk *N*1 は /tk *N*5 を /cf *V*6.*kako*。
23. /y /tk *N*3 を /cf *V*4 て /tk *N*5 を /tk *N*1 は /cf *V*6.*kako*。
24. /y /cf しっかり /tk *N*3 を /cf *V*4 て /tk *N*5 を /tk *N*1 は /cf *V*6.*kako*。
25. /y /cf 確り /tk *N*3 を /cf *V*4 て /tk *N*5 を /tk *N*1 は /cf *V*6.*kako*。
26. /y /tk *N*3 を /cf *V*4 で /tk *N*5 を /tk *N*1 は /cf *V*6.*kako*。
27. /y /cf しっかり /tk *N*3 を /cf *V*4 で /tk *N*5 を /tk *N*1 は /cf *V*6.*kako*。
28. /y /cf 確り /tk *N*3 を /cf *V*4 で /tk *N*5 を /tk *N*1 は /cf *V*6.*kako*。

3.2.2 形態素解析データの作成

パターンの形態素解析データを作成する手順を次に示す.

1. 展開前のパターンとその原文をパーサで照合
2. 照合結果を元に，各パターン要素に原文の形態素情報を割り当て
3. 変数・様相関数については，それぞれの要素に適切な形態素情報をプログラムで生成
4. 全てのパターン要素に形態素情報が割り当てられていれば，手順 8 へ進む
5. 選択要素内のパターン要素（例：(*V.kako*|*ND* をした)）は，事前に作成したデータベースを参照して形態素情報を生成
6. 補完要素内のパターン要素は，全て「*N*/は (7530)」の形式であることが分かっているので，格助詞「は」に適切な形態素情報を割り当て.
7. それでも形態素情報が割り当てられないパターン要素が存在する場合は，形態素解析データ作成不能と判断
8. パターン要素と形態素情報の対応を元に，展開後パターン全ての形態素解析データを生成
9. 文節境界情報フラグを，展開後のパターン中の離散記号により調整

手順 3 では，変数・様相関数に対して，自動的に形態素情報を作成する．これは，これらのパターン要素には，定義から形態素情報を作成できる為である．しかし，後述する文節境界情報フラグについては，パターンだけでは文節境界の判断が難しい為，前の手順で割り当てられた原文の形態素情報を参照する.

手順 5 では，選択要素内の形態素情報が割り当てられていないパターン要素に対して，データベースを参照することで形態素情報を割り当てる.

このデータベースは，256 個の「サ変型名詞変数 *ND*+ する」型の要素の形態素解析結果を収録している．収録されている「*ND*+ する」型の要素は，既に構築されている 24 万件の文型パターン辞書で用いられている選択要素内のパターン要素から抽出されている．これは，この型の要素の多くは，パターン構築時の作業により追加された為，原文に存在しないからである．なお，データベースは付録に示す.

手順 9 で調整を行う文節境界フラグは，文節の境界を示す情報である．離散記号は，パターン作成時に文節間にのみ挿入されている．そこで，離散記号の前後は文節の境となる事を利用して，フラグの調整を行う.

次に作成するパターン形態素解析データの例を示す.

- **原文の形態素解析データ**

  城を取り巻いて攻撃した。

  1. /城 (1100,{NI:449,NK:57,KR:4703u00})
  2. + を (7430)
  3. /取り巻い (2314, 取り巻く,{NY:20,KR:4104a07})
  4. + て (7630)
  5. /攻撃し (2433, 攻撃する,{NI:1762,NI:1522,NY:23,KR:3800a01})
  6. + た (7216)
  7. +。([P]0110)
  8. /nil

- **展開後パターンの形態素解析データ**

  ／ｙ／ｔｃｆｋＮ２を／ｃｆＶ３て／ｃｆＶ４．ｋａｋｏ。

  1. /／ｙ (FFFB)
  2. /／ｔｃｆｋ (FFFB)
  3. /Ｎ２ (FFF0, Ｎ,{NI:449})
  4. + を (7430)
  5. /／ｃｆ (FFFB)
  6. /Ｖ３ (FFF0, Ｖ,{NY:2002,KR:6411,KR:4104})
  7. + て (7630)
  8. /／ｃｆ (FFFB)
  9. /Ｖ４ (FFF0, Ｖ,{NY:2302,NY:2303,KR:3900,KR:3800})
  10. +．ｋａｋｏ (FFFA)
  11. +。([P]0110)
  12. /nil

3.3 パーサによる照合

パーサを用いて，全パターンの形態素解析データと全パターン間の照合を行う.

3.4 照合結果を用いた包含関係の判定

パーサの照合結果を用いて，パターン間の包含関係の判定を行う．本研究では，パターン中の記号を展開している為，展開後のパターンの照合結果を総合して判定する必要がある．その為，次の手順でパターン A とパターン B の間の関係 $A \supseteq B$ の有無を判定する．

1. パーサの照合結果から，n 個に展開された入力パターン $(B_1 \cdots B_n)$ とパターン A との照合結果 $(B_{result_1} \cdots B_{result_n})$ を取り出す．
2. 次の式より，パターン B がどれくらい A に含まれるかの割合 $(Coverage_{A \supseteq B})$ を計算する．

$$Imply_{A \supseteq B_i}(i) = \begin{cases} 1 & (\text{照合結果 } B_{result_i} \text{ にパターン } A \text{ がある}) \\ 0 & (\text{照合結果 } B_{result_i} \text{ にパターン } A \text{ がない}) \end{cases} \tag{1}$$

$$Coverage_{A \supseteq B} = \frac{1}{n} \sum_{i=1}^{n} Imply_{A \supseteq B_i}(i) \tag{2}$$

3. $Coverage_{A \supseteq B}$ が 1.0 のとき，$A \supseteq B$ とする．
   また， $0 < Coverage_{A \supseteq B} < 1.0$ のときは，展開後の一部は含まれるが，全体では包含関係に無いと言える．

3.5 下位パターンの削除

包含関係により，下位パターンと判定されたパターンを文型パターン辞書から削除する．

4　包含関係による削減の実験

4.1　実験対象

文型パターン辞書に収録されている文法・単語レベルの 122,619 パターンを実験の対象パターンとする.

4.2　実験環境

次に実験環境を示す.

- 日本語パターンパーサ Jpp(20040816kai1)
- 変数・関数定義ファイル Version 3.3.1

4.3　実験方法

前章で提案した削減方法を用いて，削減を行う.

5　実験結果

5.1　作成されたパターンの形態素解析データ

表 6 に実験対象のパターンの形態素解析データの作成結果を示す．結果より，全体の約 97% のパターンの形態素解析データを作成出来た．

表 6 パターンの形態素解析データの作成結果

入力パターン	122,619	-
作成に成功したパターン数	119,027	97.07%
作成に失敗したパターン数	3,592	2.92%
括弧の対応関係間違い	24	0.02%
原文と照合不可	1,687	1.38%
形態素データの作成失敗	85	0.07%
作成後の形態素解析データ総数		649,010
所要時間		7 時間 38 分

5.1.1 形態素解析データの作成に成功したパターン

この節では，形態素解析データの作成に成功したパターンと，その形態素解析データを示す．なお，形態素解析データは，記号の展開後のパターンに対して作成されるため，１パターンから１つ以上のデータが作成された．

■パターン例１

- **パターン**

  /ytk *N*1 を /cf *V*2 と /tk *N*3 が /cf *V*4。

- **形態素解析データ**

  ／ｙｔｋＮ１を／ｃｆＶ２と／ｔｋＮ３が／ｃｆＶ４。

  1. /／ｙｔｋ (FFFB)
  2. /Ｎ１ (FFF0, Ｎ,{NI:530})
  3. + を (7430)
  4. /／ｃｆ (FFFB)
  5. /Ｖ２ (FFF0, Ｖ,{NY:2301,NY:2205,NY:0506,NY:3201,NY:3001,NY:3002,NY:3102, NY:2601,KR:1801,KR:1603,KR:1508,KR:1800,KR:1600,KR:0400,KR:4000,KR:5502})
  6. + と (7610)
  7. /／ｔｋ (FFFB)
  8. /Ｎ３ (FFF0, Ｎ,{NI:2620,NI:2595,NI:561})
  9. + が (7410)
  10. /／ｃｆ (FFFB)
  11. /Ｖ４ (FFF0, Ｖ,{NY:2701,NY:2205,NY:2101,NY:2001,KR:8908,KR:6907})
  12. +。([P]0110)
  13. /nil

■パターン例２

- **パターン**

  /ytk *N*1 を /cf すると#2[! ずっと] /tk *N*3 が /cf *V*4。

- **形態素解析データ１**

  ／ｙｔｋＮ１を／ｃｆすると／ｔｋＮ３が／ｃｆＶ４。

  1. /／ｙｔｋ (FFFB)
  2. /Ｎ１ (FFF0, Ｎ,{NI:1985})
  3. + を (7430)
  4. /／ｃｆ (FFFB)
  5. /する (2436,{NY:16,NY:21,NY:20,NY:32,NY:5,KR:0500a33})
  6. + と (7610)
  7. /／ｔｋ (FFFB)
  8. /Ｎ３ (FFF0, Ｎ,{NI:1166,NI:926,NI:2613,NI:460})
  9. + が (7410)
  10. /／ｃｆ (FFFB)
  11. /Ｖ４ (FFF0, Ｖ,{NY:0506,NY:0502,KR:9604,KR:4803})
  12. +。([P]0110)
  13. /nil

- **形態素解析データ２**

  ／ｙｔｋＮ１を／ｃｆするとずっと／ｔｋＮ３が／ｃｆＶ４。

  1. /／ｙｔｋ (FFFB)
  2. /Ｎ１ (FFF0, Ｎ,{NI:1985})
  3. + を (7430)
  4. /／ｃｆ (FFFB)
  5. /する (2436,{NY:16,NY:21,NY:20,NY:32,NY:5,KR:0500a33})
  6. + と (7610)
  7. /ずっと (4100,{KR:7903f09})
  8. /／ｔｋ (FFFB)
  9. /Ｎ３ (FFF0, Ｎ,{NI:1166,NI:926,NI:2613,NI:460})
  10. + が (7410)
  11. /／ｃｆ (FFFB)
  12. /Ｖ４ (FFF0, Ｖ,{NY:0506,NY:0502,KR:9604,KR:4803})
  13. +。([P]0110)
  14. /nil

5.1.2 括弧の対応関係間違いにより失敗したパターン

パターン作成時に記述ミスにより，括弧が正しくネストされていなかったり，括弧が不足してたりした 24 パターンに対して，形態素解析データを作成することは出来なかった．これは，括弧を用いる記号の展開が正しく行えない為である．

以下に，記述ミスのあったパターンの一部を示す．

- /ytk *N*1 は /cf 変節して$1 /tk *N*2 に$1^{/cf 移った。
- /y #1{/tk *N*1 は, /tcfk *N*2 が }$1 /cf *V*3.*tekuru.kako^rentai* と$1$1^{/f 見える。
- /y #1{/tk *N*1 は, /tcfk #2[人目を } /cf 忍ぶ] ! *N*3.*da.kako*。

5.1.3 原文との照合が行えなかったことにより失敗したパターン

パターンの形態素解析データの作成には，原文の形態素解析データを用いている．そして，パターンと原文との対応関係を得る為に，パターンと原文との間で照合を行う．この照合が出来なかった事により作成できなかったパターンが 1,687 件あった．なお，照合が行えない理由は本研究の対象外であるため省略する．

以下に，照合が行えなかったパターンの一部を示す．

- /y </tk *N*1 は> ！ *ADV*2 /f *V*3.*kako^rentai* ！ *N*4 は</tcfk *N*5 は> ！ あくまでも /cf (*V*6^*meirei*|*V*6.*meireigo*)。
- /y </tk *N*1 は> /tcfk *N*2 を /tcfk *N*3 の /k 中に /cf *V*4(て | で) /tk *V*5 に /cf してしまった。
- /y “ *V*1 ”(という | と言う) /cf のは /cf 「(*V*2^*rentai*|*ND*2 をする)」(という | と言う) /tcfk *N*3 の /k *N*4.*da*.

5.1.4 形態素データの作成に失敗したパターン

パターン要素に対応する形態素データの作成に失敗した為，パターン全体の形態素解析データの作成が行えなかったパターンが 85 件あった．主な理由として，「*ND*+ する」型において，日本語として誤った表現があった為，生成が失敗した事が挙げられる．

以下に，パターン要素に対応する形態素データの作成に失敗したパターンの一部を示す．

- /y $1^{/tk *N*1 は } /tcfk 物を /cf 盗んで$1 /f (*V*2.*rareru.kako*|*ND*2 をしられた)。
- /y </tk *N*1 は> #1{#2[/tcfk 今朝は], /tcfk *N*3 が } /cf 鳴らず /f *V*4.*kako^rentai* ！ のは /tcfk １０時過ぎだった。
- /y </tk *N*1 は> /cf 脅したり /f (すかし | 賺かし | 賺し) たりして#2[! *ADV*3] /f (*V*4.*sase.kako*|*V*4^sase.*kako*|*ND*4 をしした)。

5.2 パーサの出力結果

パーサの出力結果を解析し，式 (1),(2) を用いて包含関係の判定を行った．5.2.1 節と 5.2.2 節で包含関係にあると判定された例と，包含関係に無いと判定された例を示す．

5.2.1 包含関係にあるパターンの例

包含関係にある日本語文型パターンとその原文を以下に示す．この例では，$Coverage_{P2 \supseteq P3} = 1.0$ であるので，$P2 \supseteq P3$ となる．

上位 *P*2: /ytk *N*1 は /tcfk *N*2 の /f ある！*N*3.*da*。
彼は身分のある人だ。

下位 *P*3: /ytk *N*1 は /tcfk 力の /f ある！*N*2.*da*。
彼は力のある政治家だ。

この例では，下位パターンの名詞「力」が上位パターンの変数 *N*2 に包含される．その結果，パターン全体に包含関係が生じている．この例のように，字面が変数に包含されることでパターンに包含関係が生じる例が，包含関係にあるパターンでは多数見られた．

また，以下に示すような，用いられる記号が異なっているが同じ表現を表すパターンも発見することが出来た．

上位 *P*4: /y $1^{/tk *N*1 は } /tcfk *N*2 を$1 /cf *V*3^*rentai* ！*N*4 が$1 /cf ある。
親は子を扶養する義務がある。

下位 *P*5: /y #1{/tk *N*1 は, /tcfk *N*2 を } /cf *V*3^*rentai* ！*N*4 が /cf ある。
彼は家族を養う義務がある。

5.2.2 包含関係に無いパターンの例

包含関係に無いパターンの例を前節と同様に以下に示す．この例では，$Coverage_{P6 \supseteq P7} = 0.5$ であるので，$P6 \not\supseteq P7$ となる．これは展開後パターンに任意要素内の “大きく” が無い場合に包含され，ある場合に包含されない為である．

上位 *P*6: /y #1{/tk *N*1 に, /tcfk *N*2 を } /cf *V*3(て | で) /cf *V*4。
凧に尾を付けて上げる。

下位 *P*7: /y #1{/tk *N*1 に, /tcfk *N*2 を } /cf 放して#3[!大きく] /f *V*4。

海に魚を放して大きく育てる。

5.3 包含関係を用いた削減結果

調査した包含関係を用い，下位パターンを削除した．この結果，パターン数は 112,767（削減率は 8.0%）となった．

5.4 出現頻度データとの関係

包含関係により削減したパターン数と，出現頻度との関係を調査した．出現頻度は，本研究で用いたパターンが適合した原文の数で，パターンと全原文との間の照合実験により求めた．２文以上に適合したパターンについて，図 2 に示す．

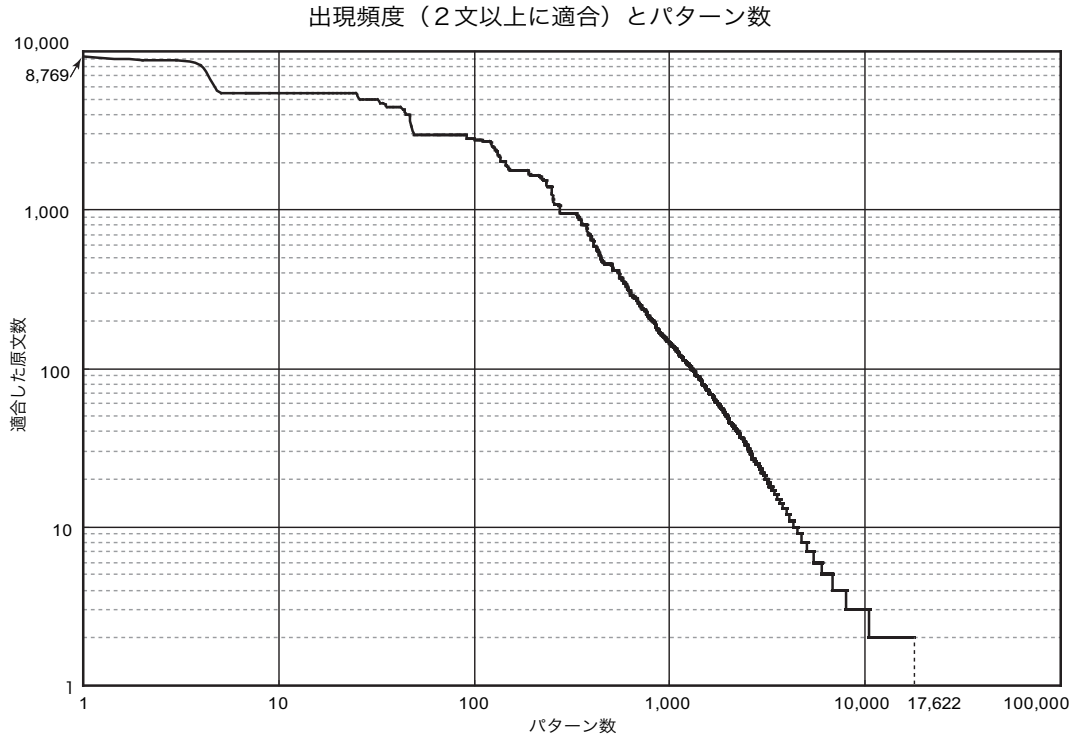

図 2 出現頻度データ

調査結果を表 7 に示す.

表 7 出現頻度データと包含関係による削減結果の関係

出現頻度	パターン数	削減パターン数
1000 文以上	275 (0.2%)	195 (70.9%)
100 文以上 1000 文未満	1,013 (0.8%)	502 (49.6%)
100 文未満 10 文以上	3,194 (2.6%)	983 (30.8%)
10 文未満 2 文以上	13,140 (10.7%)	3,255 (24.8%)
小計	17,622 (14.4%)	4,935 (28.0%)
1 文以下	104,997 (85.6%)	4,917 (4.7%)
合計	122,619 (100.0%)	9,852 (8.0%)

表 7 より，以下のことが示される.

1. 出現頻度が大きいパターンは削減率が大きい.
2. 出現頻度が小さいパターンは削減率が小さい.

5.4.1 出現頻度 1000 文以上のパターンの例

この節では，出現頻度が 1000 文以上のパターンついて例を示す.

■パターン例１

パターン: /y </tk *N*1 は> /tcfk (*ND*2 を /cf し |*V*2) て /cf (*V*3.*kako*|*ND*3 をした)。

出現頻度: 8,769

■パターン例２

パターン: /y </tk *N*1 は> /tcfk *N*2 を /cf *V*3(て | で) /cf (*V*4.*kako*|*ND*4 をした)。

出現頻度: 5,463

■パターン例３

パターン: /y </tk *N*1 は> #2[/cf *ADV*3 /f *AJ*4^*rentai*] ! *N*5 を /cf *V*6.*kako*。

出現頻度: 3,958

5.4.2 出現頻度 100 文以上 1000 文未満のパターンの例

■パターン例１

パターン:

/y #1{/tk *N*1 は, /tcfk *N*2 に } /cf *V*3(て | で) </cf *N*4 は> /cf (*V*5.*kako*|*ND*5 をした)。

出現頻度: 955

■パターン例２

パターン: /y $1^{/tk *N*1 は }#2[/tcfk 非常に]$1 /cf *AJV*3^*rentai* ! *N*4.*da*。

出現頻度: 418

■パターン例３

パターン: /y </tk *N*1 は> /tcfk *N*2 に /cf *V*3.*suitei* /f *V*4.*kako*。

出現頻度: 163

5.4.3 出現頻度 10 文以上 100 文未満のパターンの例

■パターン例１

パターン: /y #1{/tk *N*1 は, /tcfk *N*2.*da* と } </cf *N*3 は> /cf *V*4.*teiru*。

出現頻度: 92

■パターン例２
パターン: /y </tk *N*1 は> /tcfk *N*2 を /cf *V*3.*suitei* /f (*V*4^*meirei*|*V*4.*meireigo*)。
出現頻度: 45

■パターン例３
パターン: /y $1^{ あの } /k *N*1 は /cf *AJ*2^*rentai* ！*N*3 を$1 /cf *V*4.*teiru*。
出現頻度: 11

5.4.4 出現頻度 2 文以上 10 文未満のパターンの例

■パターン例１
パターン: /y </tk *N*1 は> #2[/tcfk たびたび] ！*N*3 を /cf *V*4(て | で) /cf 済みません。
出現頻度: 10

■パターン例２
パターン: /y </tk *N*1 は> /tcfk *N*2 と /cf *V*3 ながら </tk *N*4 は> /tcfk *N*5 を /cf *V*6.*kako*。
出現頻度: 5

■パターン例３
パターン: /y $1^{/tk *N*1 は } /tcfk *N*2 に /cf ある！*N*3 で$1 /cf *V*4.*teiru*。
出現頻度: 2

5.4.5 出現頻度 1 文以下のパターンの例

■パターン例１ /ytk *N*1 が /tcfk *N*2 に /cf 対し /f 不利な！*N*3 を /cf *V*4.*kako*。

■パターン例２ /y $1^{/tk *N*1 は } /cf 信じられないと /cf 言う (ように | 様に)$1 /tk *N*2 を /cf *V*3.*kako*。

■パターン例３ /y </tk *N*1 は> /tcfk *N*2 が /cf *V*3.*kako*^*rentai* ！*N*4 へ#5[/tcfk 今でも] /cf *V*6.*kako* がっている。

6　考察

6.1　包含関係の判定結果の考察

包含関係の判定結果により，以下に示すような包含関係は無いが類似度の高いパターンを発見した．この例では，$Coverage_{P2 \supseteq P7} = 0.5$ となっていた．

- *P*2: /ytk *N*1 は /tcfk *N*2 の /f ある！*N*3.*da*。
- *P*7: /y </tk *N*1 は> /tcfk 張合の /f ある！*N*2.*da*。

ここで，以下で示す新たなパターン *P*8 を作成する．

- *P*8: /y </tk *N*1 は> /tcfk *N*2 の /f ある！*N*2.*da*。

このパターンは，次の条件を満たす．

$$P8 \supseteq P2$$
$$P8 \supseteq P7$$

その結果，パターン数を少なくとも 1 つは減らすこと出来る．

従って，一部包含関係があると判定されたパターンに着目し，両者を包含する新たなパターンを作成することで，削減率の向上を図れる可能性がある．

6.2　出現頻度データとの関係の考察

出現頻度が低いパターンでは削減率が小さかった．これは，出現頻度が低いパターンは特有の表現を持つものが多い為と考えられる．

6.3 英文生成に関する考察

削除した下位パターンの日本語原文を，上位対訳パターンによって翻訳可能かどうか調査を行った．これは，日本語パターンを削除することで，対応する英語パターンも削除される．その為，削除した下位対訳パターンで翻訳可能な文が，上位対訳パターンでも翻訳可能かどうか調査する必要があると考えられる為である．

今回，上位パターンからランダムに 3 件を抽出した．そして，それぞれの下位パターンからランダムに 1 件を選び，翻訳可能か調査を行った．なお，訳語については，一番適切と思われるものを人手で選択した．

6.3.1 調査例１

上位パターン

日本語パターン: /ytk *N*1 は /tcfk *N*2 の /f ある！*N*3.*da*。

日本語原文: 彼は骨のある奴だ。

英語パターン: *N*1's *N*3 with *N*2.

英語原文: He's a man with backbone.

下位パターン

日本語パターン: /ytk *N*1 は /tcfk 力 の /f ある！*N*2.*da*。

日本語原文: 彼は力のある政治家だ。

英語パターン: *N*1 be a powerful *N*2.

英語原文: He is a powerful politician.

下位の日本語文と上位の日本語パターンとの間で照合を行い，変数と日本語単語との対応を取る．そして，日本語単語を英訳し，英語パターンに代入した結果，次の英文を得られた．

```
He's politician with power.
```

この英文は，意味的に正しくそのまま英文として使用出来る．

6.3.2 調査例２

上位パターン

日本語パターン: /y $1^{/tk *N*1 は } /tcfk *N*2 を$1 /cf *V*3(て | で)$1 /cf (*V*4.*kako*|*ND*4 をした)。

日本語原文: 彼は採用通知をもらって欣喜雀躍した。

英語パターン:　*N*1 (*V*4|V(*ND*4)).*past* to *V*3 *N*2.

英語原文:　He leapt with joy to receive the notification of employment.

下位パターン

日本語パターン:　/y $1 /tk *N*1 を /cf 見て$1^{/tk *N*2 は } /cf ぞっとした。

日本語原文:　それを見て私はぞっとした。

英語パターン:　It made me creepy.

英語原文:　*N*1 made *N*2.*obj* creepy.

この例では，以下の英文が得られた．

```
I shuddered to see it.
```

この英文は，意味的に正しくそのまま英文として使用出来る．

6.3.3　調査例３

上位パターン

日本語パターン:　/ytk *N*1 が /cf *V*2(て | で) /tk *N*3 に /cf *V*4.*kako*。

日本語原文:　氷が溶けて水になった。

英語パターン:　*N*1 *V*4 *N*3 when *N*1.*pron V*2.

英語原文:　Ice becomes water when it melts.

下位パターン

日本語パターン:　/ytk *N*1 が /cf 倒れて /tk *N*2 妨害に /cf (成っ | なっ) た。

日本語原文:　馬が倒れて交通妨害になった。

英語パターン:　The fallen *N*1 blocked *N*2 for some time.

英語原文:　The fallen horse blocked the traffic for some time.

この例では，以下の英文が得られた．

```
Horse becomes a traffic obstruction when it fall.
```

この英文は，重要な情報が間違って訳されている．

6.3.4　調査結果について

以上の 3 例より，下位パターンを削除した場合，上位の対訳パターンでは翻訳を行えなくなる日本語文が存在する可能性があることが分かった．今後，より多くの事例について，調査を行う必要があると考えられる．

6.4 削減手法の検証

6.4.1 目的

削除した下位パターンが，上位のパターンで置き換え可能かどうかを個別に調査することで，削減手法の有効性を検証する．

6.4.2 方法

削除した下位パターンの原文が，上位のパターンに適合する時に置き換え可能とする．適合の可否は，本研究で用いた文型パターンパーサを用いて判定する．

6.4.3 対象

本研究による削減の結果，上位と判定された 3,129 パターンと，下位と判定された 8,530 パターン．

6.4.4 結果

調査した結果，１組を除き全ての下位パターンの原文が，それぞれの上位パターンに適合した．適合しなかった組のパターンとその原文をそれぞれ以下に示す．

上位 WJ173142-01: /ytk *N*1 (ほど | 程) /cf *AJ2ˆrentai* ! *N*3 は /cf (無い | ない)。
LJ001279: 親ほどありがたいものは無い。
下位 WJ126067-01: /ytk *N*1 (ほど | 程) /cf *AJ2ˆrentai* ! ものは /cf ない。
LJ055262: 飢えほど恐ろしいものはない。

適合しなかった理由を調べたところ，下位パターンの原文中の名詞「もの」が，形態素解析結果では形式名詞になっていた．しかし，名詞変数 *N* に形式名詞は適合しないように定義されている．その結果,「もの」が上位パターンの変数 *N* に適合しない為，下位パターンの原文が上位パターンに適合しないと判定された．

しかし，本来「もの」は形式名詞ではなく，名詞となるべきである．そして，名詞となっている場合には，上位パターンに適合する為，原因は形態素解析の誤りだと言える．したがって，本研究で用いた削減手法では，本来は適合しないパターンを削除していないということが分かった．

6.5 英文側から見た縮退方法について

パターン間の適合から包含関係を判定する前に，英語パターンが同一であることを条件に包含関係を判定する方法を検討した．本節では，その検討結果を示す．

6.5.1 方法

以下で示す方法で調査を行った．

1. 同じ英語パターンを持つ対訳パターンのグループ化を行う．
2. グループの中から，対訳パターンを２つ以上を含むグループのみを抽出する．
3. 抽出したグループ内の日本語パターンの包含関係についての調査を，以下に示す方法で行う．
4. グループ内の日本語パターンの原文それぞれについて，グループ内の他の日本語パターンに適合するか調査を行う．
5. 調査結果から，包含関係にあるパターン対（包含関係候補）と，互いに包含関係にあるパターン対（互いに包含関係候補）を抽出する．この時の互いに包含関係とは，$A \supseteq B$ かつ $B \supseteq A$ という関係である．
6. （互いに）包含関係候補に対して，パターン間の差異を調査する．なお，この時の各パターン対は，（互いに）包含関係にあるとは断定は出来ない．2.4 節で示したように，すべての日本語文に対する調査を行っていない為である．しかし，可能性がある為「候補」としている．

6.5.2 調査対象

文型パターン辞書に収録されている文法・単語レベルの 122,619 パターンを実験の対象パターンとする．

6.5.3 グループ化の結果

同一英語パターンを持つ対訳パターンのグループ化を行い，2 つ以上の対訳パターンを持つグループの抽出を行った．その結果を表 8 に示す．

表 8 文法・単語レベルでのグループ化の結果

対象対訳パターン数	123,451
→日本語パターン数	121,284
→英語パターン数	119,849
抽出した全グループに含まれる対訳パターン数	5,918(4.8%)
→日本語パターン数	5,349(全日本語パターンの 4.4%)
→英語パターン数	2,316(全英語パターンの 1.9%)
抽出されなかった対訳パターン数	117,533(95.2%)

グループ化を行った結果，対訳パターン全体の 95.2% が，縮退の対象外となった．これにより，同一の英語パターンを持つ対訳パターンは少ないということが分かる．

以下にグループ化された対訳パターンの例を示す．

■例１

英語パターン: *N*1 be *AJ*2.
グループ内対訳パターン数: 18
対訳パターン１:
日本語パターン: /ytk *N*1 とは /cf *AJV*2ˆ*rentai*！ものだ。
日本語原文: 人生とは空虚なものだ。
英語原文: Life is empty.
対訳パターン２:
日本語パターン: /ytk *N*1 の /f 言う！ことは /cf *AJV*2。
日本語原文: 君の言うことは本当だ。
英語原文: You are truthful.

■例２

英語パターン:　*N*1 *V*3.*past* to *N*2.

対訳パターン１:

日本語パターン:　/y $1^{/tk *N*1 は } /tcfk *N*2 を /cf 目指して$1 /cf *V*3.*kako*。

日本語原文:　開拓者たちは西を目指して進んだ。

英語原文:　The pioneers advanced to the west.

対訳パターン２:

日本語パターン:　/y </tk *N*1 は> /tcfk *N*2 が /cf (付く | つく | 点く | 附く | 傅く)(まで | 迄) /cf *V*3.*kako*。

日本語原文:　決着がつくまで戦った。

英語原文:　We fought to the finish.

6.5.4　包含関係候補の調査結果

（互いに）包含関係候補のパターン対に対して，パターン間の差異を調査した結果を表 6.5.4 に示す．なお，調査した差異は，位置指定記号のみ，離散記号と位置指定記号のみ，語尾関数のみ，様相関数のみ，離散記号と語尾関数のみ，離散記号と様相関数のみ，の 6 種類で，いずれも簡単な差異となっている．

表 9 パターン間の差異の調査結果

		互いに包含関係候補		包含関係候補	
パターン対の数		123		214	
(日本語パターン数)		(208)		(334)	
調査した差異	離散記号のみ	15	12.2%	1	0.5%
	位置指定記号 ($1) のみ	37	30.1%	0	0.0%
	離散と位置指定のみ	5	4.1%	0	0.0%
	語尾関数のみ	0	0.0%	0	0.0%
	様相関数のみ	2	1.6%	0	0.0%
	離散と語尾のみ	0	0.0%	0	0.0%
	離散と様相のみ	0	0.0%	0	0.0%
	計	59	48.0%	1	0.5%

結果より，今回調査した差異を持つ互いに包含関係候補のパターン対は 48% と半分程で，残りの候補はそれ以外の差異を持つことが分かった．また，包含関係候補のパターン対では，ほとんど調査した差異を持たなかった．

以下に包含関係候補の例を示す．

■位置指定記号のみが異なっている例

英語パターン:　*N*1 be *AJ*3 like *N*1.*poss N*2.

対訳パターン１:

日本語パターン:　/y $1^{/tk *N*1 は } /tcfk *N*2 に /cf 似て $1 /f *AJV*3。

日本語原文:　彼は父親に似てひょうきんだ。

英語原文:　He is good-humored like his father.

対訳パターン２:

日本語パターン:　/y $1^{/tk *N*1 は } /tcfk *N*2 に $1 /cf 似て $1 /f *AJV*3。

日本語原文:　彼は父親に似て勤勉だ。

英語原文:　He is diligent like his father.

この例では，位置指定記号 ($*n*) のみが，日本語パターン間で異なっていて，対訳パターン２の方が１箇所多い．

6.5.5　結論

本章では，英語パターンが同一という条件で包含関係を判定する方法を検討した．しかし，英語パターンによりグループ化を行った結果，4.4% の日本語パターンのみが調査対象となった．さらに，調査対象中の（互いに）包含関係候補のパターン間の差異も，簡単な物は半分以下であることが分かった．

以上より，本章で検討した方法では，調査対象の少ないという問題がある．また，日本語原文が別の日本語パターンに適合することから，包含関係にあると断定出来ない問題もある．その為，別の方法を検討する必要があると考える．

7　おわりに

本研究では，日本語文型パターン間の包含関係に着目して，文型パターン数の削減方法について提案した．また，その方法を用いて大規模文型パターン辞書のパターン数削減を試みた．

その結果，文法・単語レベルのパターン辞書 (122,619 パターン) において，8.0%(9,852 パターン) を削除することが出来た．また，出現頻度が高いパターンは削減率が大きいが，出現頻度が低いパターンは削減率が小さかった．なお，本研究で提案した削減手法は誤っていない事も分かった．しかし，削減を行うことで，翻訳が行えなくなる可能性があることも分かった．

今後，包含関係には無いが，類似度の高いパターンを調査し，両者を含む新たなパターンを作成する手法が考えられる．また，削減が出来なかったパターンについては，縮退自体の必要性の有無も検討していく．

謝辞

最後に，本研究においてご指導頂きました，鳥取大学工学部知能情報工学科計算機工学 C 研究室の池原悟教授, 村上仁一助教授，徳久雅人助手, そして研究室の皆様に深く感謝します.

また，参考にさせて頂いた文献の著者の方々に対して感謝します.

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, 言語処理学会論文誌 Vol.11, No.3, pp.69-95, 2004

[2] 池原 悟，村本 奈央，徳久 雅人，村上 仁一，宮崎 正弘，佐良木 昌:機械翻訳のための日英文型パターン記述言語の設計, 電子情報通信学会技術研究報告,TL2002-48, pp.1-6, 2003

[3] 徳久 雅人, 村上 仁一, 池原 悟: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

[4] James Allen:Natural Language Understanding(2nd Edition), The Benjamin/Cummings Publishing Company, Inc., pp.101-106, 1994

付録

1. 追加・変更した変数・関数定義オートマトンファイル
    - vara.implycheck - A 型変数定義ファイル
    - vars.implycheck - S 型変数定義ファイル
    - risan.implycheck - 離散記号定義ファイル
    - funcg.implycheck - 語尾関数定義ファイル
    - funcy.implycheck - 様相関数定義ファイル
    - def_function.implycheck - 関数定義設定ファイル
2.「サ変型名詞変数 ND+ する」型の形態素解析情報データベース
3. 実験に用いた削減方法を実装したブロック図
4. 発見した包含関係の上位パターン (下位パターン 2 件以上)

付録１ 追加・変更した変数・関数定義オートマトンファイル

付録２ 「サ変型名詞変数 ND+ する」型の形態素解析情報データベース

付録３ 実験に用いた削減方法を実装したブロック図

付録４ 発見した包含関係にあるパターン (上位パターンのみ)