多義解消のための構造規則の生成方法と日本語名詞句への適用

池 原　悟[†]　中井　慎司[†]　村上　仁一[†]

自然言語処理では，処理の過程で，さまざまな解釈の曖昧さが生じる．この曖昧さを解消するのに必要な知識を記述するため，対象とする表現を部分的な表現の組に還元せず，一体として捉える方法として，言語表現とその解釈の関係を変数とクラスの組からなる構造規則として表現し，学習用標本から半自動的に収集する方法を提案した．この方法は，パターン化された表現の変数部分を表すのに文法属性体系と意味属性体系を使用しており，N 個の変数を持つ表現パターンに対して，一次元規則から N 次元規則までの規則と字面からなる例外規則を合わせて $N+1$ 種類の構造規則が順に生成される点，また，各規則は，その生成過程において，各属性の意味的な包含関係を用いて容易に汎化される点に特徴がある．本方式を「A の B の C」の型の名詞句に対する名詞間の係り受け解析規則の生成に適用した結果では，変数部分を意味属性で表現した構造規則の場合，1 万件の学習事例から，一次元規則 198 件，二次元規則 1480 件，三次元規則 136 件が得られ，それを使用した係り受け解析では，約 86% の解析精度が得られることが分かった．また，変数部分を文法属性で表した規則と意味属性で表した規則を併用する場合は，解析精度は，1 ~ 2% 向上することが分かった．この値は，2 名詞間の結合強度に還元して評価する方法 (72%) より約 15% 高い．この種の名詞句では，人間でも係り先の判定に迷うような事例が 10% 近く存在することを考慮すると，得られた規則の精度は，人間の解析能力にかなり近い値と言える．

キーワード:　知識獲得，構造規則，構造知識，名詞句解析，係り受け規則，文法属性，意味属性，意味解析，自動学習，多義解消

Automatic Knowledge Acquisition for Disambiguation and Its' Application to Japanese Noun Phrases

Satoru Ikehara [†] , Shinji Nakai [†] and Jin'ichi Murakami [†]

In order to represent the knowledge for resolving the syntactical and semantic ambiguities, structural rules and their generalization methods were proposed. In this method, the structural rules are composed of structure definition part and class definition part. The former is written by the set of any of almighty symbol, syntactic attributes, semantic attributes and word itself. From the view point of the number of parameters used for defining the expression structures, the rules are classified into one-dimensional rules, two-dimensional rules and so on, and automatically generated in this order from examples. Prominent feature of this method is in generalization methods. The generated rules are furthermore generalized based on the upper to lower relation of semantic attributes or syntactic attributes to reduce the number of rules without decreasing the performance.This method was applied to generate the dependency rules for Japanese expressions of "A no B no C" which are known as the most popular noun phrases which have hard ambiguities to be resolved. As the

result, it was found that structural rules can easily be obtained by this method. The experimental results showed that the dependency relations can be determined at the accuracy of 86% by the rules obtained by this method. This rate is so low compared to human ability for this kind of ambiguous noun phrases.

KeyWords: *Knowledge Acquisition, Semantic Rule, Semantic Analysis, Noun Phrase, Leaning, Disambiguation, Syntactic Attribute, Semantic Attribute, Dependency Analysis*

1　はじめに

自然言語処理では，機械翻訳システムの研究開発を中心に，過去 10 年以上にわたって多大な投資が行われ，言語解析アルゴリズムなど，大きく発展してきた (田中穂積 1989; 長尾真 1996; 田中穂積 1999) が，解析の過程で発生する表現構造と意味に関する解釈の曖昧性の問題は，依然として大きな問題となっている．日本語の構文解析では，特に，述語間の係り受け関係の曖昧さ (白井ほか 1995) と並列構造の識別 (黒橋, 長尾 1994) が問題とされているが，名詞句 (冨浦ほか 1995; 菊池，白井 2000) や複合語 (小林ほか 1996) の構造の曖昧さも大きな問題である．英語では，前置詞句の係り先の曖昧さ (隅田ほか 1994) などがクローズアップされている．また，機械翻訳では，訳文品質低下の最大の原因は，動詞や名詞の訳語の不適切さにある (麻野間，中岩 1999) とも言われており，訳語選択の問題 (桐澤ほか 1999) は，解決の急がれる問題の一つとなっている．

ところで，このような解釈の曖昧性が発生する原因は，解析アルゴリズムにあるのではなく，解析に使用される情報や知識の不足にある (Ikehara 1996)．曖昧性は，解析の途中で生じた複数の解釈の候補の中から，正しい解釈が選択できないことであるから，選択に必要な情報がある場合は発生しない．これに対して，解析アルゴリズムは，与えられた情報を使用して解釈を決定する手順であるから，優れたアルゴリズムでも，不足している情報を補うことは不可能である．従って，曖昧性の問題を解決するには，不足する情報を見極め，それが，与えられた表現から得られないときは，辞書や知識ベースとして外部から補うことが必要である．

ここで，与えられた表現の意味を決定する問題について考えると，要素合成法の原理に従えば，表現の意味は，それを構成する単語から合成されることになる．すなわち，辞書によって各単語の語義が与えられると，それらの組み合わせによって表現の意味が決定できることになる．このような観点からの研究としては，単語に対して詳細な語彙情報を用意し，それを組み合わせて表現の意味解釈を生成する生成意味論の方法 (Pustejovsky 1995)，オントロジーをベースとした知識処理の方法 (Nierenburg et al. 1992; 武田ほか 1995)，言語処理のための意味表現の研究 (内海ほか 1993) などがある．しかし，現実の言語表現では，個々の単語の役割と意味は，与えられた表現の中で，その単語が占める位置に依存して決定しなければならない場合も多く，

† 鳥取大学工学部知能情報工学科, Department of Information and Knowledge Engineering, Faculty of Engineering, Tottori University

そのため，表現構造に関する知識や情報が必要となる．事例から情報を得て処理を進める方法 (長尾 1984; 佐藤 1992;Sumita and Iida 1992)，単語の共起関係の情報を使用する方法 (小林ほか 1996; 麻野間，中岩 1999;Piva Alves, et.al. 1998)，さらには，単語の共起関係をパターン化する方法 (池原ほか 1993; 宇津呂ほか 1993;Almuallim et.al. 1994b; 池原ほか 1997) などは，いずれも表現の構造に関する情報を使用している．

このように，表現構造に関する情報は，曖昧性解消のための重要な手がかりと言えるが，解析に先立ってこれらの情報を網羅的に収集することは容易でない．通常，自然言語において，語彙に関する情報は，高々，数十万語が対象と見られるのに対して，その組み合わせである表現の場合は，ほぼ無限と言える．また，表現構造には，広い範囲で一般化できるものや，個別的で汎用化の困難なものなどがあり，ばらつきが大きい．

そこで，本論文では，コーパスなどの言語データから曖昧性解消に必要な表現構造の知識を収集するための方法の一つとして，言語表現とその解釈の関係を変数とクラスの組からなる構造規則として表現し，学習用標本から半自動的に収集する方法を提案する[1] ．本方式では，対象とする表現を字面による文字列部分と変数部分 (他の単語に置き換え可能な部分で，制約条件を単語の属性で記述する) からなるパターンで表わし，そのとき使用された変数の組によって表現構造を定義する．

ところで，このような構造規則によって多様な言語表現をカバーするには，大量の標本が必要であり，必要とされる規則数も大きいと予想される．また，多数の規則を相互矛盾なく定義するには，文法属性だけでなく，粒度のきめ細かな (属性数の多い) 意味属性の体系が必要になると予想される．

ここで，従来の学習技術との関係をみると，種々の帰納的学習の方法が提案されてきたが，学習事例数，意味属性数，生成される規則数が共に大きい問題では計算が難しい．大規模な木構造からなる意味属性を使用する点から見ると，本論文の問題は，従来の格フレーム学習 (Almuallim, et al. 1994b) と同種の問題であり，(Haussler 1988) の方法の適用が期待される．しかし，この方法は，学習事例数の増大に弱く，数千件以上の学習事例では実用的でない．また，事例数に強い方法としては，(Quinlan 1993) の決定木学習の方法が知られているが，この方法は，木構造で表現されるような属性間の背景知識を使用する場合には適用できない．この問題を解決する方法として，木構造をフラットな属性列にエンコーディングするなど，いくつかの方法 (Quinlan 1993;Almuallin et al. 1994b; アルモアリムほか 1997) が提案されているが，いずれも，事例数，属性数，規則数が共に大きい問題に対する適用は容易でない[2] ．

1 本論文では，従来の還元論的な方法で解決できない曖昧性を解消することを目指しており，曖昧さが問題となる表現をいくつかの部分的な表現の組に分解することはせず，一体として扱う．

2 Quinlan の方法では次元数 N 個分の意味属性の木を組み合わせて一つの新しい木を作るのに対して，(Almuallim et al. 1994a) の方法は，意味属性の木を次元数× 意味属性数のビット列に展開する．これに対して，(アルモアリム 1997) の方法は，エンコーディングをせずに，予め，事例を属性木に「流し」，ノード上に事例情報を蓄えておくことにより，直接計算を可能とするものである．本論文のように，事例数，属性数，規則数が共に大きい問題の場合，計算量は，Quinlan の方法の方が小さい．しかし，この方法は，意味属性のレベルに対して粒度がバランスしていないときは，精

そこで，本論文では，実用性を重視する観点から新しい方法を提案する．本方式の構造規則は，構造定義に使用された変数の数に着目して，一次元規則，二次元規則などの次元規則に分類されるが，解析精度を落とさず，汎用的な構造規則から順に生成することを考え，一次元規則から順に生成する．また，得られた各次元の構造規則に対し，木構造で表現された文法属性と意味属性の意味的包含関係を利用した自動的な汎化の方法を示す[3]．

本論文では，提案した方法を日本語名詞句に適用してその効果を確認する．具体的には，「A の B の C」の形の名詞句の事例から名詞 A の係り先を決定するための解析規則を生成し，生成した規則を解析に使用してその適用範囲 (カバー率) と解析正解率を求める．

2　構造規則の記述方法

2.1　構造規則の基本型

自然言語処理において発生する解釈の曖昧さの種類はさまざまである．それを解消するために必要な情報も，文脈情報や常識，世界知識や専門知識などさまざまで，すべての種類の曖昧さを解決できるような情報をあらかじめ網羅的に準備することは困難である．

ここで，従来の自然言語処理を見ると，表現の構造に関する知識は，文法知識として意味から分離し，表現の意味は，文法規則に従って語彙情報から合成する方法が一般的であった．しかし，実際の自然言語では，表現の構造とその意味を一体化して扱うことの必要な場合も多い．従来の言語解析で発生する解釈の曖昧さの多くは，むしろ，このような構造と意味に関する知識をあらかじめ準備することによって解決できる可能性がある．そこで，本論文では，このような知識を「構造規則」として，その形式を定義し，収集する方法を考える．

さて，このような「構造規則」は，一般に，表現構造を定義する部分とその解釈を定義する部分から構成できる．このうち，言語表現の構造を記述する方法としては，木構造，リスト構造，意味ネットワークなど様々な方法がある．言語表現の構造的，意味的多義について考えると，文字列や品詞の並びから見た限りでは，類似または同等と思われるような表現が，統語的もしくは意味的に複数の解釈を持つことが問題である．そこで，対象とする表現構造を，言語表現そのものに近く，理解しやすい表現として，文字もしくは記号の連鎖からなる一次元的なパターンとして表現する．すなわち，構造規則の対象とする表現は，キーとなる文字列部分 (「固定部」と称す) と形式的に他の単語や表現に置き換えられる部分 (「変数部」または，単に「変数」とも言う) から構成されるパターンで表現する[4]．但し，前者は，字面で記述され，後者は，通常，記号で記述される．

度が保証されない．

3 本論文の方法では，必ずしも，必要最小限の規則のセットが生成されるとは限らない．最近の計算機の記憶容量を考え，無理なく実装可能なルール数に収斂すればよいと考える．

4 例えば，「＜名詞＞から＜名詞＞への＜名詞＞」の表現パターンでは，2 つの文字列「からの」，「への」を「固定部」，3 つの＜名詞＞を「変数部」という．

ここでは，このような構造規則は，曖昧性が問題となる表現の種類毎に収集されるものとする．例えば，名詞句「東京の叔父の息子」では，名詞「東京」の係り先の解釈が問題となるが，これは，「A の B の C」(A ,B ,C は，いずれも名詞) の形の名詞句における名詞 A の係り先の問題として構造規則を用意する．「美しい私の娘」における形容詞「美しい」の係り先の問題では，「形容詞 + A の B」の表現構造における形容詞の係り先多義の問題として別の構造規則を作成する．また，名詞句「A の B」の英語への翻訳規則の場合，「山の頂上」(top of the mountain)，「すべての学生」(all of the students)，「私の友人」(my friend)，「嵐の夜」(stormy night)，「京都の寺」(temple in Kyoto) などのように，名詞 A と名詞 B の組み合わせの違いなどによって，訳し方に多義が存在するが，これも表現構造と解釈の関係として定義される．すなわち，「東京の叔父の息子」と「美しい私の娘」に対する係り受け解析では，異なる規則集合を作成する．また，名詞句「A の B」の翻訳の問題でも別の規則集合を作成する．

このように，構造規則の対象とする表現を，曖昧性の問題となる表現の種類毎にパターン化した場合，表現パターン内の定数部分は省略し，変数部分のみによって表現構造を定義しても問題はないから，パターン化された表現の中から変数部分だけを取り出し，表現の構造を変数の組 (tuple) として表現する．

以上から，本論文では，多義解消のための構造規則の基本形を下記の通りとする．

$$(A_1, A_2, A_3, \ldots, A_N : C) \tag{1}$$

但し ,A_i : 変数部 ,N : 変数の数 ,C : クラス

以下では，構造規則のうち，$A_1, A_2, A_3, \ldots, A_N$ の部分を「構造定義部」，C の部分を「クラス定義部」と呼ぶ．

2.2　表現構造とクラスの記述

ところで，ほぼ無数とも言える言語表現をなるべく少ない構造規則でカバーするには，汎用性の高い構造規則を生成することが望まれが，一方，多彩な言語表現をカバーするためには，個別的な表現に対する規則も記述できる必要がある．そこで，汎用性の程度に応じて柔軟に規則を記述するため，「構造定義部」の変数 A_i は，下記に示す 4 種類の言葉もしくは記号のいずれかで記述するものとする．

$$A_i = \begin{cases} *: & \text{オールマイティ (無指定)} \\ \text{文法属性}: & \text{品詞，活用行，活用形など} \\ \text{意味属性}: & \text{単語の意味的用法を表す言葉} \\ \text{字面}: & \text{標準表記された原文文字列} \end{cases} \tag{2}$$

上記の変数は，「オールマイティ」，「文法属性」，「意味属性」，「字面」の順に適用範囲が広いと考えられる．すなわち，「オールマイティ」は，制約条件のないことを意味しており，最も汎用性

が高い．「文法属性」，「意味属性」では，使用する文法体系の違いなどによって，種々の分類法が考えられるが，通常，言語解析では「文法属性」は，数 10 程度に分類されるのに対して，「意味属性」は，数百から数千種類に分類される．これに対して，字面情報は，単語の数で見ても 10 万種類以上となり，それで定義された規則は汎用性に乏しい規則となるが，言語表現には，慣用句など字面指定によって解釈の決まるような表現も多数存在する．

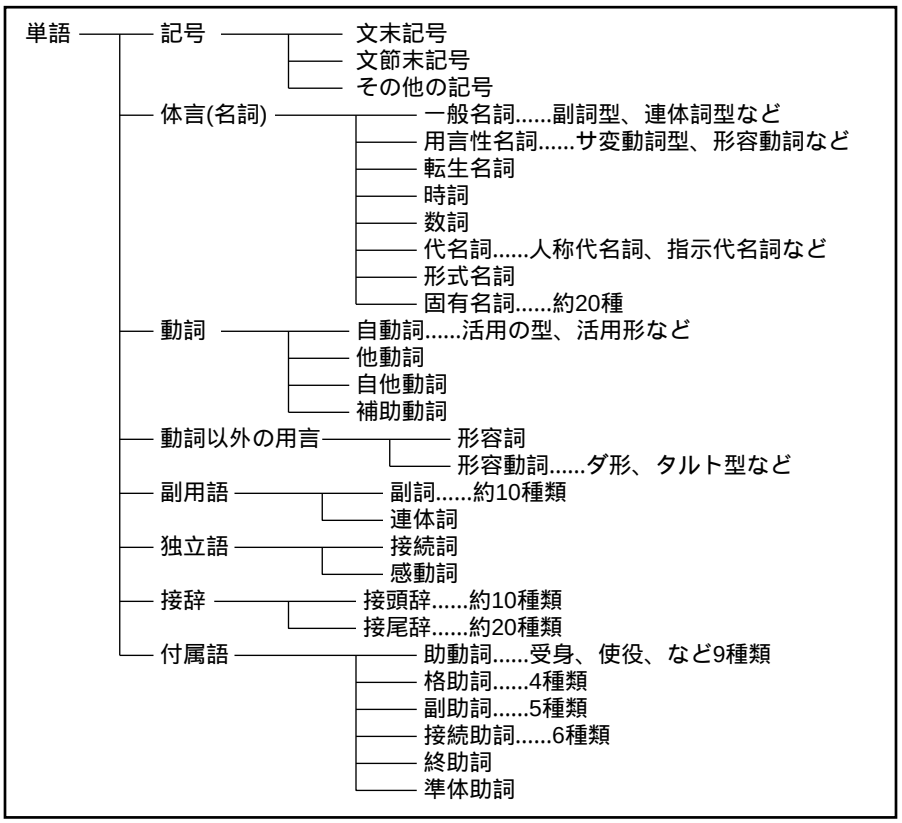

図 1　文法属性体系の例 (日英機械翻訳システム ALT-J/E の場合)

本論文では，「文法属性」として，日英機械翻訳システム ALT-J/E で使用されている「文法属性体系」(池原ほか 1987; 宮崎ほか 1995) を使用し，「意味属性」としては，「日本語語彙大系」(池原ほか 1997) で定義された「単語意味属性体系」を使用する．文法属性体系と単語意味属性体系の一部を，それぞれ，図 1，図 2 に示す．

次に，式 (1) のクラス定義部のクラス C は解釈を示す記号であり，構造規則の種類に応じて，統語的，意味的解釈を与える記号として使用される．例えば，前述の「A の B の C」の形の名詞句では，名詞 A の係り先が名詞 B の場合と名詞 C の場合の 2 種類の解釈の可能性があるから，これを区別するには，構造規則では，$C = \{b$ 係り ,c 係り $\}$ とすればよい．また，「形容詞

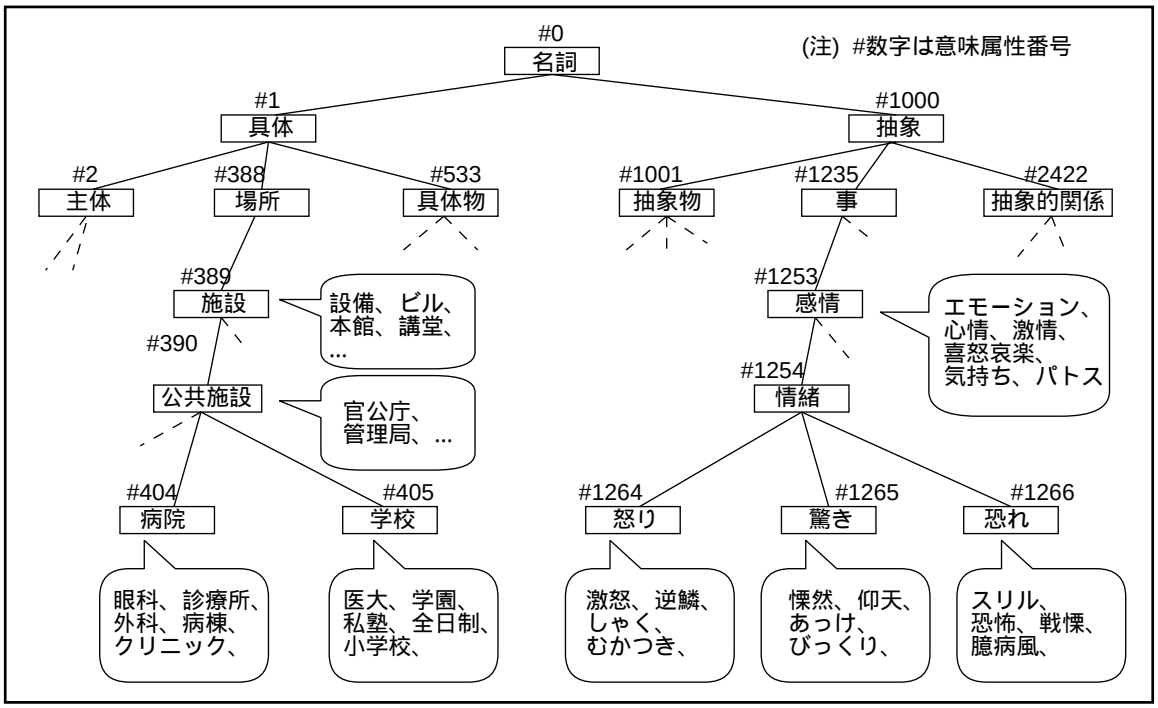

図 **2**　一般名詞意味属性体型の例 (「日本語語彙体系」の場合)

+ A の B」では，$C = \{a$ 係り ,b 係り $\}$ となる．また，「A の B」の英訳規則の場合は，$C = \{$of 型 ,in 型，所有代名詞型 ,$\ldots\}$ のようになる．

3　構造規則生成の基本的な考え方

本章では，コーパスなどから得られた事例を対象に，構造規則を発見し汎化するための考え方について述べる．その方法は，表現構造が文法属性によって定義された構造規則 (簡単のため，「文法属性規則」と言う) と意味属性によって定義された構造規則 (同様，「意味属性規則」と言う) のいずれの場合も同様であるが，前者は，従来から検討されており，人手による標本分析で比較的容易に作成できるので，以下では，後者の場合を中心に述べる[5]　．

3.1　標本の収集と学習用事例の作成

以下では，コーパスなどの原文から抽出した表現の文字列とそれに対する解釈 (クラス) からなるペアを「標本」，その集合を「標本集合」と呼び S_0 で表す．また，標本を式 (1) の形式の書き換えたものを「事例」，その集合を「事例集合」と呼び，S_1 で表す．

5 表現が意味属性と文法属性の両者によって定義される構造規則の作成については，3.3(2) で触れる．

例として，「A の B の C」の形の名詞句に対する係り受け規則の場合の例を以下に示す．

名詞句の標本集合 S_0 : 私の母の友達 (私 → 母)
名詞句の事例集合 S_1 : (私，母，友達 : B)　(3)
但し，B は，名詞 A(「私」) が名詞 B(「母」) に係ることを示す

ここで，集合 S_1 の中のすべての変数の値 (字面) を対応する意味属性で置き換えて得られた事例集合を S_2 とする．上記の名詞句の事例では，S_2 の要素として，

名詞句の事例集合 S_2 : (#8「自称」#80「母」#125「友人」: B)　(4)
但し #n は，意味属性番号，「」内は，意味属性名を表す．

が得られる．以下では，簡単のため，これを，(#8 #80 #125 : B) または，(8 ,80 ,125 : B) のように記す．

3.2　規則の次元分類とその発見

(1) 特徴空間と構造規則の次元

前章で示した構造規則の記述方法に従えば，学習用事例 S_2 も構造規則と同じ (1) 式の形式で表現されるから，S_2 の各要素は，それ自体，構造規則と見なすことができる．事例からのこのような規則生成では，帰納的推論の方法 (長尾 1988) の適用が考えられる．

そこで，式 (1) の構造定義部で指定された N 個の変数に対して，各変数を基底とする N 次のベクトル空間 (「特徴空間」とも言う) を考えると，各事例は，特徴空間上の点に対応するから，構造規則の生成は，この特徴空間内で，同一のクラスに属す部分空間を切り出す問題となる．

この種の問題は，特徴空間が線形である場合，クラスタ分析もしくはクラスタリングの問題 (安西 1989; 浅野，江島 1996;Witten & Frank 1999) としても良く知られており，多変量解析，情報検索などの分野で研究されている．しかし，式 (1) の構造定義部で与えられる各基底はいずれも非線形である[6] ため，計算は簡単でない．基底が is-a 関係で結ばれた木構造となる場合については，(Haussler 1988) の方法があるが，学習事例が多い場合は，適用困難である．また，事例数の大きい問題への適用を狙った方法として，意味属性の木構造をエンコーディングした後，既存の計算プログラム C4.5 を使用する方法など (アルモアリムほか 1997) もあるが，学習事例，基底数，(生成される) 規則数が共に大きい場合は，やはり計算困難である．そこで，本論文では，変数の数に着目して構造規則を次元に分類し，意味属性間の包含関係に着目して規則を生成する方法を考える．

さて，特徴空間上，同一のクラスに属す点の集合に対して構造規則は定義される．すなわち，

6 例えば，意味属性で表現された変数の場合，変数の値は，意味属性番号である．意味属性番号間では加法定理は成り立たず，特徴空間上で事例間の距離もしくはノルムを定義することができない．

構造規則は，一般に，N 次空間上の点，もしくは，特定の領域に対応する．これに対して，N 個の変数のうち K 個の変数がオールマイティ「$*$」で表現された規則は，K 次元だけ縮退された規則となり，$N-K$ 次元の空間内のベクトルで表現されるが，N 次元空間で見れば，$N-K$ 次元の立方体に対応する規則であり，この立方体内に属す事例に適用される．

例えば，三次元の構造規則 $(A_1, A_2, A_3 : C)$ において，A_1, A_2, A_3 の何れかを「$*$」で置き換えた規則 (例えば，$(*, A_2, A_3 : C)$) は，三次元空間上の線に対応する規則となり，2 変数を「$*$」で置き換えた規則 (例えば，$(*, A_2, * : C)$) は，三次元空間上の面に対応する規則となる．

以上から，構造規則をそれが定義される特性空間の次元に従って，一次元規則から N 次元規則までの N 種類に分類する．

(2) 構造規則発見の方法

事例集合 S_2 から構造規則を生成するための基本的な考え方について述べる．

対象とする表現の意味は，その前後の文脈に依存せず，与えられた表現だけで決定できると仮定[7] すると，対象とする表現とそのクラスが 1 対 1 の関係を持つ．すなわち，標本集合 S_0 と事例集合 S_1 において，同一の表現構造が異なるクラスに属す要素はない．

しかし，字面を意味属性に置き換えて得られた集合 S_2 では，構造定義部とクラス定義部が 1 対 1 に対応する場合と，1 対 1 には対応せず，同一の構造に対して異なる複数のクラスが対応する場合が存在すると考えられる[8] ．このうち，前者は，構造定義部で定義された表現の解釈は一意に決定できることを意味しているから，このような部分集合から「意味属性規則」が生成できる[9] ．すなわち，

(1) 集合 S_2 の要素を構造定義部の等しい要素毎に分類する．分類されたグループ内の要素が，いずれも同一のクラスを持つとき，そのグループから一つの構造規則が生成できる．生成された規則は，該当するグループの要素数が多いほど，信頼性が高い．

(2) 分類されたグループ内の要素のクラスが一致しないときは，そのグループからは構造規則は生成できない．その場合，「意味属性規則」は存在しないと判断できるから，もとの S_2 の要素 (字面表記) を構造規則 (「字面規則」と呼ぶ) とする．

このうち，(1) で得られる構造規則は，構造定義に使用された意味属性相互の包含関係を使用すれば，さらにグループ化することができて，より汎用的な規則の生成が期待できる．これに対して，(2) で得られた規則は，一般化の困難な表現，すなわち慣用表現に類する表現の規則

7 文脈依存のある場合は，クラスが一意に決定できる範囲まで，構造定義の範囲を拡大すればよい．

8 要するに，字面レベルでは，表現とクラスの関係は，1 対 1 であるが，意味属性に置き換えた表現では，必ずしも 1 対 1 にならない．

9 得られる規則の精度は，部分集合に属す事例数に依存する．そのため，後に述べるように目標とする精度に応じた閾値を設け，その値以上の事例数の部分集合から規則を生成する．

であると推定されるから，汎化の対象外となる[10] ．

3.3　規則生成の順序と汎化

(1) 構造規則の生成順序と適用順序

すでに述べた一次元規則から N 次元規則までの規則では，次元の小さい規則ほど制約条件が少なく，汎用性が高い．また，そのような規則は高速に適用できるから，規則生成においては，一次元規則から順に生成する[11] ．このとき，規則生成で使用された学習事例を後の規則生成で再び使用するか否かが問題となる．

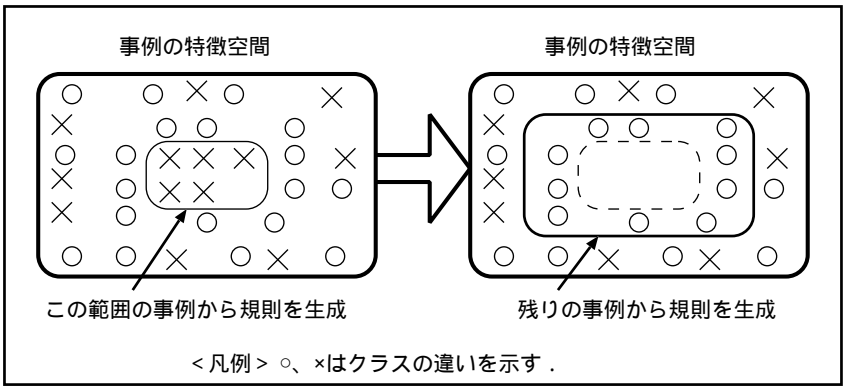

図 3　構造規則汎化の方法

ここで，言語解析では，得られた規則をその生成順に適用することとすると，後に生成された規則が，それ以前に生成された規則の適用範囲に適用されることはないから，後に生成された規則が先に適用される規則の適用範囲を含んでいても何ら問題は発生しない．そこで，規則生成では，図 3 の例に示すように，生成に使用した事例は，事例集合から削除し，後の規則生成は，残された事例を対象に進める．このように，事例の特徴空間から，ある部分的な空間から規則を抽出した後，その空間に含まれる事例を消去すると，その後の規則生成では，より広い範囲での汎化が期待できる[12] ．以下では，この方法を「逐次型生成[13] 」と呼ぶ．

10 表現構造を定義する N 個の変数のすべてが単語字面で記述された規則が生成されるが，その中には，一部の単語字面を意味属性に置き換えてよい (汎化できる) 規則が含まれている可能性がある．そのような規則は，意味属性から文法属性への書き換えの場合 (3.3(2) 参照) と同様，人手によって発見し，汎化するものとする．

11 明確な証明はないが，従来，単純な規則と複雑な規則が混在しているときは，単純な規則の方を優先する方が良さそうだ (「オッカムのかみそり」の原則) と言われている．C4.5 でも，決定木の生成順序の決定で同様の作戦が用いられている．

12 すでに述べたように，特徴空間は非線形であるので，汎化を進めるに当たって，空間を単純に拡張併合することができない．汎化の具体的方法は 4 章で述べる．

13 これに対して，すべての事例を対象に，各次元を規則を生成する方法を「同時型生成」と呼ぶ．「同時型生成」については，後述する．なお，これらの方法は，事例呈示法の分類から見るといずれも「同時提示 (simutaneous presentation) 」に属すもので，段階的に学習事例を提示する「逐次提示 (incremental presentation) 」ではない．

以上から，構造規則生成の手順をまとめると，以下の通りとなる．

＜構造規則生成の手順＞

(1)　事例集合 S_1 から，事例集合 S_2 を作成する．S_2 は，S_1 の各標本内の変数部分の単語をその単語の属す意味属性番号に置き換えたものである．

(2)　事例集合 S_2 から，一次元規則の集合 R_1 を生成し汎化する[14] ．そのとき，規則生成に使用した事例は，S_2 集合から削除する．

(3)　上記で残った事例 S_2 から，二次元規則の集合 R_2 を生成し，汎化する．そのとき，規則生成に使用した学習事例は，S_2 集合から削除する．

(4)　以下同様にして，N 次元までの規則集合 $R_i(1 \leq i \leq N)$ を生成する．

(5)　以上の結果，残された学習用事例 S_2 の要素に対して，その元となった事例標本の集合を R_e とする．(R_e は，事例そのものであるが，同時に構造規則でもある．)

なお，各次元規則の生成と汎化の方法は，次章で述べる．以上で得られる構造規則の種類は，以下の通りである．

(1)「意味属性規則」の集合　$\sum_{i=1}^{N} R_i$　(5)

(2)「字面規則」の集合　R_e　(6)

(2) 文法属性を使用した規則への汎化

以上の方法で生成された構造規則を対象に，構造定義部の意味属性を文法属性に置き換えてよい規則の組を探して，それらを文法属性による規則に置き換える．置き換えられた規則では，次元やタイプの異なる複数の構造規則が縮退されるため，適用順序の情報が失われる．従って，書き換え後の規則は，適用順序に依存しない独立した規則である必要がある．そこで，ここでは，書き換えの可否は，人手により判断するものとする．

さて，前項までで得られる構造規則は，変数部分がいずれも「オールマイティ」,「意味属性」,「字面」の何れかで記述された規則である．ここで,「意味属性」で記述された規則の組を「転生名詞」,「時詞」,「形式名詞」など，より汎用な「文法属性」で記述された規則に汎化することを考える．一つの「文法属性」に複数の「意味属性」が対応することに着目し，各文法属性毎に，それと対応した意味属性を持つ規則を集め，該当する意味属性の部分を文法属性で置き換えた規則を作成する．新しい規則の作成では，後に述べる意味属性規則の汎化と同様，着目する意味属性以外の要素の同一性に注意する必要がある．また，書き換え後の規則の独立性

14 全体が N 次元からなる構造規則の場合，一次元規則は，指定される意味属性の位置によって N タイプの規則に分類される．i 次元規則の場合，タイプの数は ${}_N\mathrm{C}_\mathrm{i}$ となる．総合的な品質が，生成する規則のタイプの順序に依存するかどうかは，実際に生成した規則によって判断する．後に述べるように，本論文の例題では，同次元内で構造規則の生成順序を変えても，得られる規則全体の精度は変わらなかった．

を保証するため，木構造上，置き換え対象となる規則の適用領域内に他の規則が存在していないことを確認する必要がある[15]．

このようにして得られた文法規則を，文法属性の上下関係に着目して，さらに汎化する場合も同様である[16]．

なお，一般に，構造規則において，表現構造定義部は，字面，意味属性，文法属性などの混在する形式で記述できるから，上記の置き換えは，可能な変数のみを対象とすればよい．

4　構造規則の生成手順

各次元の「意味属性規則」を生成し，汎化する方法について述べる．

4.1　一次元規則の生成

(1) 一次元規則の発見

さて，一次元規則を発見する方法について述べる．まず，表現構造を規定する N 個の変数に対応して，N 個の意味属性体系の木を用意する．用意した各木のノードに「事例数リスト $(n_1, n_2, \ldots, n_K)$」を対応させる．但し，n_i は，該当するノードの意味属性を持つクラス i の事例数で，K はクラスの数である．例えば，m 番目の変数に対応する意味属性体系の木の $\#j$ 番目のノードの場合，n_i は，事例集合 S_2 の中で，m 番目の変数の値が $\#j$ である要素の数を表す．以下では，このようにして得られた意味属性体系の木を「意味属性数の木」と呼ぶ．

図 4 に，クラス数 $K = 2$ の場合について，構造定義部の m 番目の変数に対応した「意味属性数の木」の例を示す．

ここで，必要十分の標本データから「意味属性数の木」が求められているとし，m 番目の属性の木の $\#j$ 番目のノードに付与された「事例数リスト」$(n_1, n_2, \ldots, n_K)$ の各数値について考える．各クラスの事例数を示す K 個の数値のうち，i 番目のクラスの事例数 n_i を除くすべての事例数が 0 であるとすると，このノードの表す事例，すなわち，m 番目の変数が $\#j$ 番目の意味属性であるような事例は，他の変数 (m 番目の変数以外) の値とは無関係に，すべてクラス i に属すことになる．従って，このノードから，一次元の構造規則 $(*, \ldots, *, j, *, \ldots, *\colon i)$ を生成する．但し，$\#j$ は，先頭から m 番目の変数の値である．このとき，規則生成の対象となったノードの「事例数リスト」の値は，すべてゼロにリセットする．

「意味属性数の木」のノードで，事例数リストが 0 でない要素を 2 つ以上を持つノードでは，

15 以上の処理は機械化することも可能であるが，文法属性は数が少ないため，人手作業とした．

16 意味属性数に比べて文法属性数は数が少なく，書き換え可能な文法属性は予想がつきやすいので，いずれの場合も，狙いを定めて書き換えの可能性を調べるとよい．但し，文法属性の場合，属性数が少ないことから，汎化による規則数削減の効果は，あまり期待できない．「文法属性規則」は，人手による標本分析で比較的容易に作成できるため，既存のシステムでは，すでに，構造解析規則として使用されている場合が多いと予想されるのに対して，従来の構文解析で解決できなかったような曖昧性を意味解析によって解消するためには，当面，意味属性による規則の収集ができることが要請される．

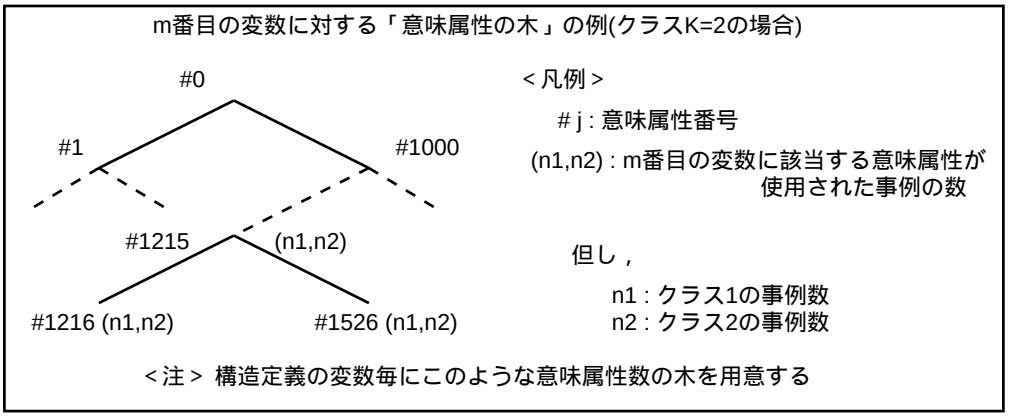

図 4 「意味属性の木」と一次元規則の発見

一次元規則は存在しないから，そのまま残しておき，後に述べるような二次元以上の規則生成を試みる．なお，すべての要素が 0 であるようなノードでは，構造規則を生成しない．

(2) 一次元規則の汎化

構造規則は，精度を失わない限り，汎用性が高く，規則数の少ない方がよい．意味属性体系上の上位の意味属性の語の性質は，下位の意味属性の語に伝搬することに着目すると，構造規則において，ある意味属性が指定されているとき，その意味属性の配下の意味属性を持つ語はすべて指定条件を満たすものと解釈される[17] ．

そこで，「意味属性数の木」のなかで，一次元規則の生成で使用されたノードに着目する．このノードから生成された一次元規則のクラスが C であり，かつ，その下位ノードのいずれからも同じクラス C の構造規則が生成されるとする．ただし，下位ノードには，対応する事例が存在せず，すべての要素がゼロとなる事例数リストは存在しても良いが，ゼロでないような要素が複数存在する事例数リストはないものとする．このとき，下位ノードから得られる規則は，着目したノードの規則で代表することができる．

汎化は，このように「事例数リスト」が上位ノードに畳み込めるようなノードを発見し，そのノードから生成された規則を削除することによって行われる．具体的には，下位ノードから汎化を開始し，順次，上位ノードに向かって汎化を進める．一度，汎化の結果得られたノードも，上記の条件を満たす限り，さらに上位ノードに縮退される．

図 5 に，クラス数 $K = 2$ の場合の例を示す．図中，(a) では，$\#165$ のノードの配下に，

17 前項の方法で生成された構造規則は，規則の定義で使用された意味属性に直属する単語の範囲にのみ適用され，その意味属性の配下にある意味属性では，別の構造規則が使用される．従って，上位の意味属性からなる規則が下位の意味属性の規則を包含するとはいえない．しかし，言語解析に適用する規則を選択するときは，表現に使用された単語の意味属性から順に上位の意味属性を辿り，最も近くの意味属性で記述された構造規則を選択すれば，それが適用すべき規則である．

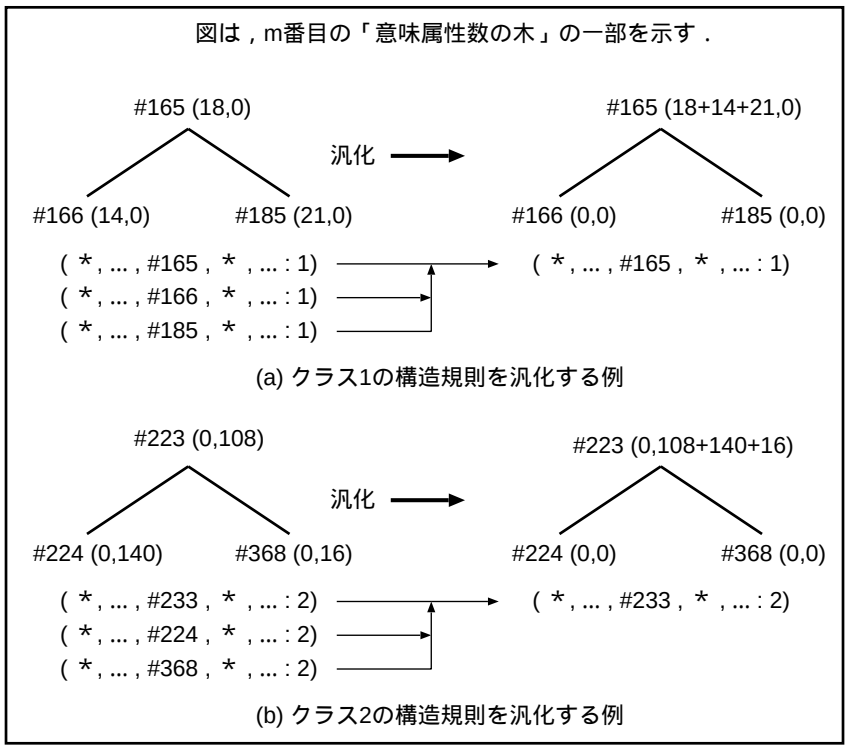

図 **5**　一次元規則の汎化の方法 $(\text{クラス数 } K = 2 \text{ の場合})$

$\#166$，$\#185$ の 2 つのノードがあるが，それらに属す事例 $(\text{それぞれ，}14 \text{ 件，}21 \text{ 件})$ は，いずれも，$\#165$ ノードの事例 (18 件) と同じく，クラス 1 の事例であるので，上位ノードに畳み込まれ，$\#165$ ノードの事例数リストは，$(53, 0)$ となる．このとき，2 つの下位ノートの事例数リストの値は，0 にクリアされる．図中の (b) は，クラス 2 の場合の例で，以下同様である．

4.2　二次元以上の規則の生成

(1) 二次元規則の発見

二次元規則では，表現を規定する N 個の変数のうち，2 個の変数の値が与えられるとクラスが決定されるから，得られる規則は，指定される変数の組み合わせによって，$N(N-1)/2$ 組に分類される．以下では，そのうちの任意の一組の規則について考える．

さて，対象とする表現が，m_1 番目の変数と m_2 番目の変数で定義されるような二次元の構造規則 $(m_1 \leq m_2 \text{ とする})$ を抽出する．m_1 番目と m_2 番目の「意味属性数の木」の情報から，行番号，列番号をそれぞれの「属性数の木」のノード番号 (意味属性番号) とし，要素を「事例数リスト」$(n_1, n_2, \ldots, n_K)$ とする二次元配列を作成する．但し，n_i は，変数 m_1，m_2 の値が，

それぞれ行番号，列番号で示される意味属性であるような事例のうち，クラスが i である事例の数を表す．図 6 に $K = 2$ の場合の例を示す．

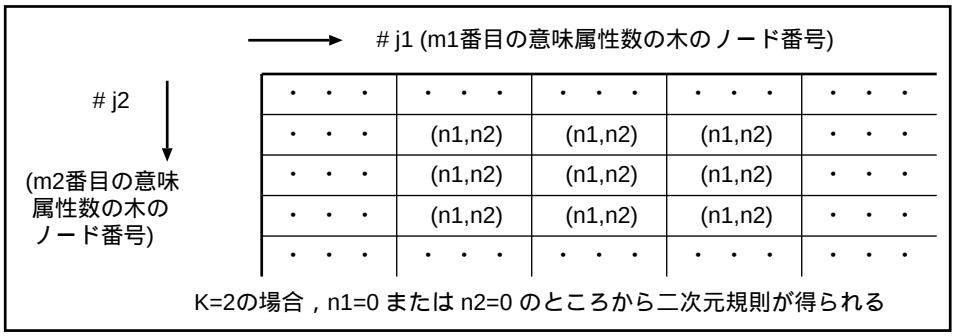

図 **6**　二次元規則の抽出

一次元規則の場合とほぼ同様，二次元規則は，この二次元配列から求められる．その方法は以下の通りである．

二次元配列の要素に示された K 個の事例数のうち，どれか一つを除くすべての数値が 0 であるような要素を考える．このような要素は，該当する変数の位置に，配列上の行と列で表される意味属性の語が使用された事例では，例外なくそのクラスが一意に定まっていることを示している．このことにより二次元規則は容易に抽出できる．

例えば，いま，$\#j_1$ 行 $\#j_2$ 列の位置の要素 $(n_1, n_2, \ldots, n_K)$ の値が，$(0, 0, \ldots, n_i, 0, \ldots, 0)$ であるとすると，下式の二次元規則が得られる．

$$(*, \ldots, \#j_1, *, \ldots, \#j_2, *, \ldots; i) \tag{7}$$

但し $n_i \neq 0$，また $\#j_1$ $\#j_2$は意味属性番号で，変数リストの先頭より，
それぞれ ,m_1番目 ,m_2番目に位置する．

なお，一次元規則の場合と同様，規則生成後，当該ノードの事例数リストの値は，すべてゼロにリセットされる．

(2) 二次元規則の汎化

二次元規則は，前述の二次元配列を使用して汎化する．ただし，行と列で表される意味属性の上下関係 (包含関係) の情報については，意味属性体系を参照する．この場合，表現を指定する変数が二種類あるため，二方向での汎化が必要な点を除けば，汎化の方法は，一次元規則の場合と同様である．

図 7 に，クラス数 $K = 2$ の場合の汎化の例を示す．図では，m_1 番目の変数の値が意味属性 $\#125$ 配下にあり，m_2 番目の変数の値が意味属性 $\#522$ の配下にある二次元規則を汎化してい

る．初めに，行方向の汎化で，9 つの構造規則が 3 つの構造規則に縮退され，次に，列方向の汎化で，3 つ構造規則が 1 つに縮退されるから，全体では，9 つの構造規則が最終的に 1 つに縮退される．

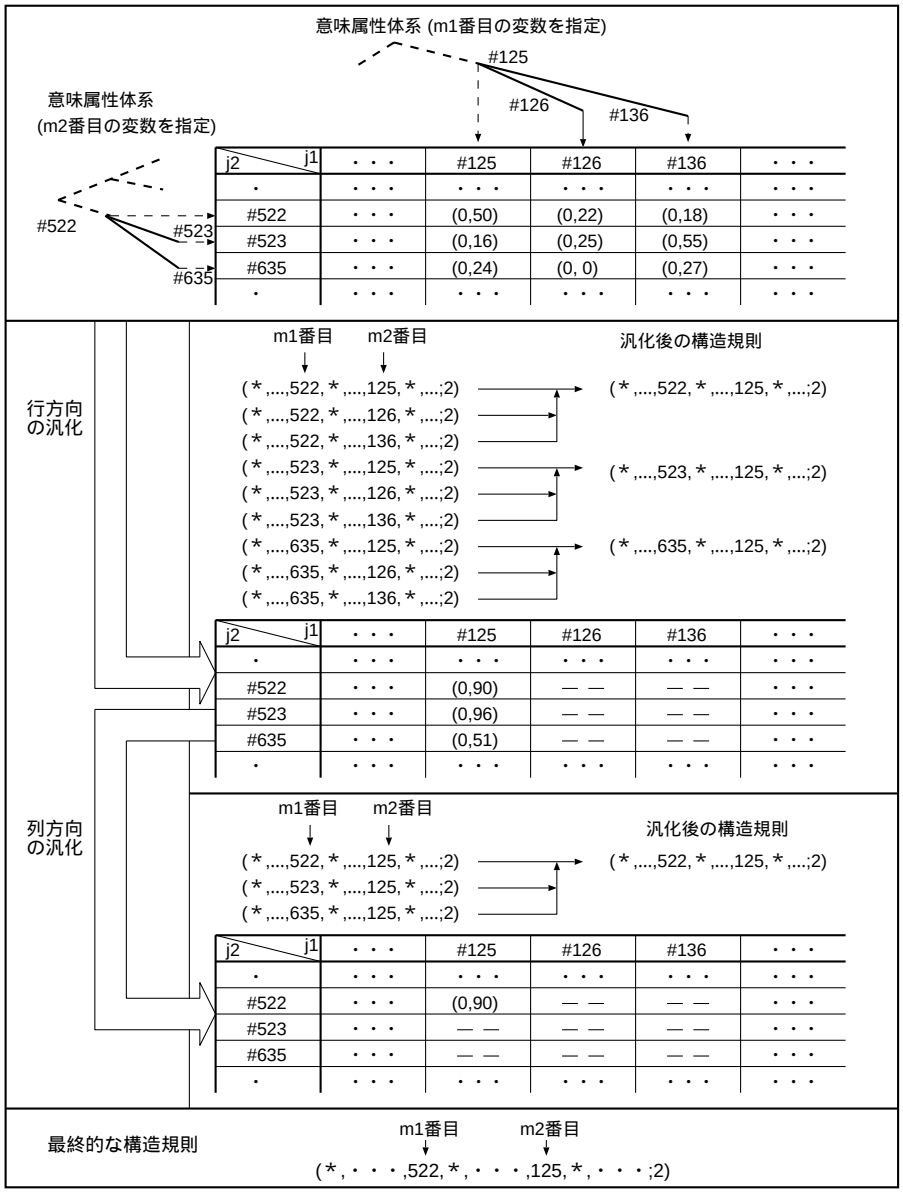

図 7　二次元規則の汎化の例

この例では，行方向と列方向のいずれから汎化しても結果は変わらないが，場合によっては，汎化の順序によって縮退できる規則数に差が生じることが考えられる．従って，規則数の減少

を図るため，双方向の汎化の結果を比べて，縮退できる規則数の大きい方を採用する．

(3) 三次元以上の規則の抽出と汎化

二次元規則が事例数リストの二次元配列から得られたのと同様，m 次元規則は，事例数リストの m 次元配列から求められる．規則を求める方法とそれを汎化する方法は，次元数の違いを除けば，二次元規則の場合と同様である．

4.3 規則生成の閾値について

(1) 規則生成対象となるノードの事例数

生成される規則の信頼性の観点から見たとき，事例数の少ないノードから生成された規則は信頼性に乏しい．従って，規則抽出は，ある程度以上の事例数を持つノードからに絞ることが望まれる．ここで，ある確率分布に従ってランダムに発生する事象を考えると，着目する現象の出現頻度はポアソン分布に従い，その信頼性は，その事象の出現回数の絶対値のみで決まる．そこで，規則生成のための事例数の閾値として，ξ 件を設定する．すなわち，構造規則の生成においては，「事例数リスト」の数値の和 $(n_1 + n_2 + \ldots)$ が ξ 以上となるノードを対象に，規則生成を試みる．

(2) 規則の精度を保証するための閾値

前節で示した各次元の規則生成では，事例の特徴空間上，例外なく同一のクラスの事例からなる空間からのみ構造規則を抽出し，それを汎化している．しかしこの方法では，ごくまれに発生する例外のため，規則が生成できないような場合が心配される．そこで，汎化の範囲をより拡大するため，この条件をゆるめ，規則生成の対象となる事例数に対して閾値 $\gamma\%$ を設ける．

すなわち，規則生成において，クラス C に対する構造規則を生成するとき，クラス C 以外のクラスの事例が $\gamma\%$ 以下であるノードの範囲まで特徴空間を広げて構造規則を生成する．汎化においても同様の基準を使用するものとする[18]．

閾値の値は，目標とする解析精度に依存して設定する必要がある．すなわち，閾値 $\gamma\%$ 内の事例から得られた規則の場合，それを使用した解析では，最大 $\gamma\%$ の誤りが生じることが予想されるから，目標とする解析精度を $\eta\%$ とするときは，閾値は，$\gamma \leq (100 - \eta)\%$ となるように設定する必要がある．

18 決定木学習における枝刈りの方法としては，決定木を最後まで成長させてから，決定木上の一つ一つのノードの価値を調べ，悪影響の大きいノードから順に削除する方法と得られた決定木から規則集を生成して，精度の悪い規則を削除する方法がある (Mitechell 1997) が，ここでは，計算量を削減するため，2 つの単純な指標に基づき，規則生成の段階で生成の可否を判断する方法とした．

5　名詞句への適用例

前章で提案した構造規則の生成法を，係り受け関係に多義を持つ名詞句として典型的な「の型名詞句」に適用し，係り受け解析のための「意味属性規則」を生成する．また，生成された規則を解析に使用して，その精度を評価する．

名詞句の意味については，すでに人手によって詳細な意味分類 (島津ほか 1986) が行われてきた．また，本章で対象とする「の型名詞句」についても，人手による標本分析の結果として種々のヒューリスティックスが提案されており，名詞間の接続強度と用例を併用した解析方法の研究 (江尻, 宮崎 1998) では，9 割前後の係り受け精度が達成されている．しかし，計算機による解析では，解析精度の問題など，まだ多くの課題を残している．係り受け解析としては，コーパスに基づく方法として，単語の共起情報を用いて係り先を決定する方法 (佐々木 1995)，複合名詞に意味クラスの共起情報を用いて係り先が決定される確率を求める方法 (小林 1996) などがある．名詞句翻訳では，生成語彙論の立場から，語彙情報によって英語表現を生成する方法 (菊池，白井 2000) もあるが，精度は不明である．また，大量の対訳例の中から意味的に類似した表現を発見し，翻訳結果を得る方法では，収集される用例は，通常，スパースであり，適切な用例がないときは，結果は保証されないことが問題であった．

本論文の方法は，名詞句の持つ意味的な構造に着目した受け規則が生成できるので，比較的少ない事例から相対的にカバー範囲の広い構造規則が生成できると期待される．

5.1　意味属性規則の生成

5.1.1　対象とする名詞句と実験の条件

(1) 対象とする名詞句とその曖昧性

さて，2 つの助詞「の」と 3 つの名詞 A ,B ,C から構成された「A の B の C」の形の名詞句を考える．ただし，記号 A ,B ,C は名詞の出現順序をも表すものとする．以下，この型の名詞句を単に「の型名詞句」という．

日本語では，一般に，表現要素間に後方修飾の原則があることに注意すると，「の型名詞句」では，名詞 B の係り先は名詞 C に特定されるため，先頭の名詞 A について，以下の 2 通りの係り受け解釈が存在する．但し，$\alpha \rightarrow \beta$ は，α が β に係ることを示す．

1)　$A \rightarrow B$　$(\&B \rightarrow C)$ の場合
　　例)「私の母 の名前」,「浴室の脱衣場 の壁」
2)　$A \rightarrow C$　$(\&B \rightarrow C)$ の場合
　　例)「私の 昔の 友達」,「東京の 数学の 教師」

以下では，簡単のため，1) を「b 係り」，2) を「c 係り」と呼ぶ．

(2) 解析規則の記述形式

名詞句「A の B の C」の構造を (X, Y, X) で表す．ただし，X ,Y ,Z は，それぞれ，名詞 A ,B ,C の属す意味属性の番号とする．次に，この構造の名詞句に対する係り受け規則を式 (1) の記法に従って，$(X, Y, X : D)$ で表す．ただし，D は係り受けのタイプで，$D = b$ は前方係り受け，$D = c$ は後方係り受けを表すものとする．この規則を次元によって分類すると図 8 のようになる．

一次元規則の種類 : $(X,\ *,\ * : D), (\ *, Y,\ * : D), (\ *,\ *, Z : D)$

二次元規則の種類 : $(X, Y,\ * : D), (\ *, Y, Z : D), (X,\ *, Z : D)$

三次元規則の種類 : $(X, Y, Z : D)$

但し，X：名詞 A の意味属性番号，　Y：名詞 B の意味属性番号，

Z：名詞 C の意味属性番号，

D：係り受けの種類 (b：前方係り，c：後方係り)

図 **8**　生成する係り受け規則の種類

ここで，オールマイティ記号「*」は，名詞の意味属性のノード番号 0 に対応する．すなわち，ノード番号 0 は，ルートノードで，すべての名詞を表すから，係り受け規則上は，意味的制約のないことを意味する．以下では，図 8 の三種類，7 タイプの構造規則を生成する．

(3) 実験対象と実験の手順

まず，小説 100 冊 (新潮文庫) を対象に，形態素解析プログラム ALT-JAWS (NTT 1996) を使用して「の型名詞句」を抽出する．そのうちの 1 万件について，人手によって係り先を決定し，名詞句の事例集合 S_1 を作成する．次に，「日本語意味属性体系」(池原ほか 1997) に定義された「単語意味属性体系」を参照して，各名詞句標本の変数部分に相当する名詞を意味属性番号に置き換え[19] ，事例集合 S_2 を作成する．ただし，名詞は複数の意味的な用法を有する場合が多いが，ここでは，各名詞の名詞句内での意味を考え，単一の意味属性に置き換える．

以下，このようにして得られた事例集合 S_2 から構造規則を生成し，得られた規則を実際の名詞句の係り受け解析に適用する．解析結果を，あらかじめ人手で決定しておいた正解と比較して解析規則の解析精度を求める．

実験は，10 回の cross-validation 法で行う．すなわち，まず，事例標本を規則生成用の $9{,}000$ 件と解析実験用の $1{,}000$ 件に分け，前者から，すでに述べた方法で解析規則を生成する．得ら

19 多くの名詞は，意味的に複数の用法を持つため，複数の意味属性に属すが，それが使用された表現では，意味的用法は，通常 1 つである．従って，与えられた名詞句では，各名詞が，どの意味で使用されているかを判定し，一つの意味属性を決定する必要がある．

れた規則を後者の標本の解析に適用して解析精度を求める．この手順を 10 回繰り返して得られた結果を平均して，生成される規則数とその精度を求める．図 9 に実験の手順を示す．

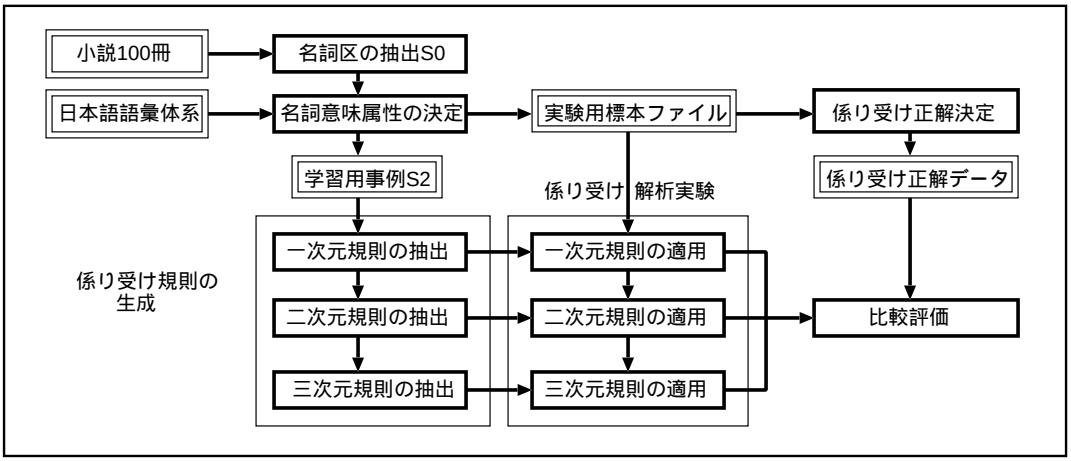

図 9　実験の手順

5.1.2 実験結果

(1) 名詞句に使用された名詞とその意味属性

「の型名詞句」の標本 1 万件で使用されている名詞 A ,B ,C の意味属性を集計した結果を表 1 に示す．この表で，「深さ」の欄は，意味属性体系上，該当する意味属性が，トップノードから何番目の深さにあるかを示す．数値が大きくなるにつれて，該当する名詞の意味の粒度が小さくなる．

表 1　名詞と意味属性の関係

深さ	意味属性の総数	名詞 A		名詞 B		名詞 C	
		異なり属性数	適用度数	異なり属性数	適用度数	異なり属性数	適用度数
0	1	1	9	0	0	1	3
1	2	0	0	0	0	0	0
2	6	2	11	0	0	2	292
3	21	12	82	13	102	12	216
4	106	56	1,580	57	843	60	871
5	256	129	1,069	161	1,973	156	3,096
6	536	219	2,051	263	2,099	244	2,354
7	827	272	2,689	382	2,275	349	1,473
8	687	262	2,020	354	2,020	307	1,272
9	211	78	474	99	653	86	403
10	40	6	7	9	24	8	16
11	16	6	8	7	11	4	4
計	2,709	1,063	10,000	1,345	10,000	1,231	10,000

これより，以下のことが分かる．

1) 使用された意味属性は，いずれの名詞の場合も意味属性全体の半分以下であり，意味的に見て，名詞の種類全体をカバーする範囲にはない．
2) 使用された意味属性当たりの標本数は，平均 $7 \sim 10$ 件である．
3) 名詞 C は，名詞 A ,B に比べて，浅い意味属性の名詞，すなわち，粒度の大きい名詞が使用される傾向がある．

ここで，木構造上のノード (意味属性) に対応した構造規則が生成されることを考えると，2) より，ほぼ $7 \sim 10$ 件程度の事例から 1 構造規則が生成されると見込まれる[20] ．その場合，規則生成の対象となるノードが反事例を持たないとすると，得られる規則の精度は，約 80% 以上となることが期待できる．また，3) は，名詞句「A の B の C」において，名詞 C の意味は，名詞 A または名詞 B によって限定されることの多いことを物語っている．

(2) 実験結果

構造規則生成実験によって生成された規則数とそれを使用した名詞句の解析実験の結果をまとめて表 2 に示す．表中，「デフォールト規則」の欄は，解析実験において，適用できる規則が存在しない事例はすべて，「b 係り」と解釈したことを示す．規則生成に使用する事例数の閾値 ξ は，一次元規則と二次元規則の生成では 2，三次元規則の生成では，1 とし，例外事例に関する閾値 γ は 0 とした．なお，一次元規則，二次元規則の生成において，それぞれの 3 タイプの構造規則の生成順序を変えても得られた構造規則全体の解析精度は変わらなかった．

< 規則抽出結果 >

表 2 から，構造規則生成の結果について以下のことが分かる．

1) 得られた意味属性規則数は，全体で $1{,}815$ 件である．この規則は，$9{,}000$ 件の事例から得られているから，平均してみれば，5 事例から 1 規則得られたことになる．
2) 各次元規則の中で，二次元規則の数が最も多く，80% 以上を占めている．
3) 三次元規則は，136 件で，他の次元の規則に比べて最小である．

このうち，2)，3) から，この種の名詞句は，三つの名詞のうち二つの意味関係によって係り受け関係が決まることが多く，三つの名詞すべてに依存する場合は少ないことが分かる[21] ．

20 構造規則は，$1 \sim 3$ 個の意味属性によって記述されるから，実際は，規則当たりの事例数は，もう少し小さくなると予想される．

21 例えば，二次元規則では，2 つの名詞の意味属性によって係り受け関係が決定できるが，このことは，全体を 2 つの名

表 2　生成された構造規則の数と精度

番号*	構造規則種別	規則のタイプ	得られた規則数		規則適用回数	累積の事例数	精度
1	一次元規則	$(X,*,*:D)$	89.6		800 (8.0%)	800 (8.0%)	92.0%
2		$(*,Y,*:D)$	81.5	小計	591 (5.9%)	1,391 (13.9%)	88.7%
3		$(*,*,Z:D)$	27.3	198	253 (2.5%)	1,644 (16.4%)	89.3%
4	二次元規則	$(X,Y,*:D)$	888.8		4,187 (41.9%)	5,831 (58.3%)	90.6%
5		$(*,Y,Z:D)$	355.1	小計	1,917 (19.2%)	7,748 (77.5%)	89.0%
6		$(X,*,Z:D)$	236.2	1480	782 (7.8%)	8,530 (85.3%)	77.5%
7	三次元規則	$(X,Y,Z:D)$	136		453 (4.5%)	8,983 (89.8%)	68.4%
— —	合計	— — —	1,815		8,983	8,983 (89.8%)	88.0%
- - - -	デフォールト規則	b 係り	- - - -		1,017 (10.2%)	10,000 (100%)	66.6%
合計		- - - - -	1,815		10,000 (100%)	10,000 (100%)	85.8%

$(*.)$ 規則生成の順序，及び解析での規則適用の順序

＜解析実験結果＞

次に，上記の解析規則を使用した名詞句解析実験の結果から，以下のことが観察される．

1) 1 万件の事例から全体で，カバー率 89.8% の規則が得られる．また，規則のカバーする範囲の解析正解率は，平均 88.0% である．

2) 一次元規則と二次元規則の精度は，ほぼ，同程度であるのに対して，三次元規則の精度は低い．

3) 適用できる規則の存在しない事例は，すべて「b 係り」と解釈した結果，全体の正解率は，85.8% である．

このうち 1) は，従来の人手で作成された規則 (宍倉，宮崎 1995; 江尻, 宮崎 1998) の精度 (90% 前後) より若干低いが，本論文と同一の名詞句に対する従来の要素合成法的な解析規則 (中井ほか 1998) より，かなり優れている[22] ．2) は，一次元規則と二次元規則を生成する段階で，事例の多くが使用済みとなり，三次元規則の生成に使用された事例が少ないためと考えられる．また，3) の値は，人間でも判断に迷う事例が 10% 程度存在する[23] ことを考えると，かなり良い値と解釈される．

5.2　文法属性規則の作成と意味属性規則との併用

前節で使用した名詞句の事例から「文法属性規則」を作成する場合，また，それを「意味属性規則」と併用する場合についての例を示す．

詞の組に分けてもよいことを意味しない．何次元のどのタイプの規則が生成されるかは名詞句で使用される名詞の位置と意味によって決まるからである．

22 名詞句「A の B の C」を「A の B」と「A の C」に分け，それぞれの名詞の結合強度を意味属性間の共起頻度で評価することによって，名詞 A の係り先を決定する方法である．本論文と同一の意味属性 (2,700) を使用した結果では，使用する意味属性を 81 種に絞ったとき，全体の係り受け精度は最大 (72%) になる．この結果から，名詞句を分割する還元論的な方法に比べて，要素分割をしない whole 方式の効果はかなり大きいこと (この問題の場合，$86\% - 72\% = 14\%$) が推定される．

23 実験では，各事例に対して三人の人手によって係り先を付与し，多数決によって正解を決定した．

(1) 文法属性規則の生成

「A の B の C」の形の名詞句では，変数部分はすべて名詞であるため，図 1 の文法属性体系の中の体言 (名詞) の部分で示される文法属性を使用した構造規則を考える．具体的には，使用する文法体系は図 10 の通りとする．

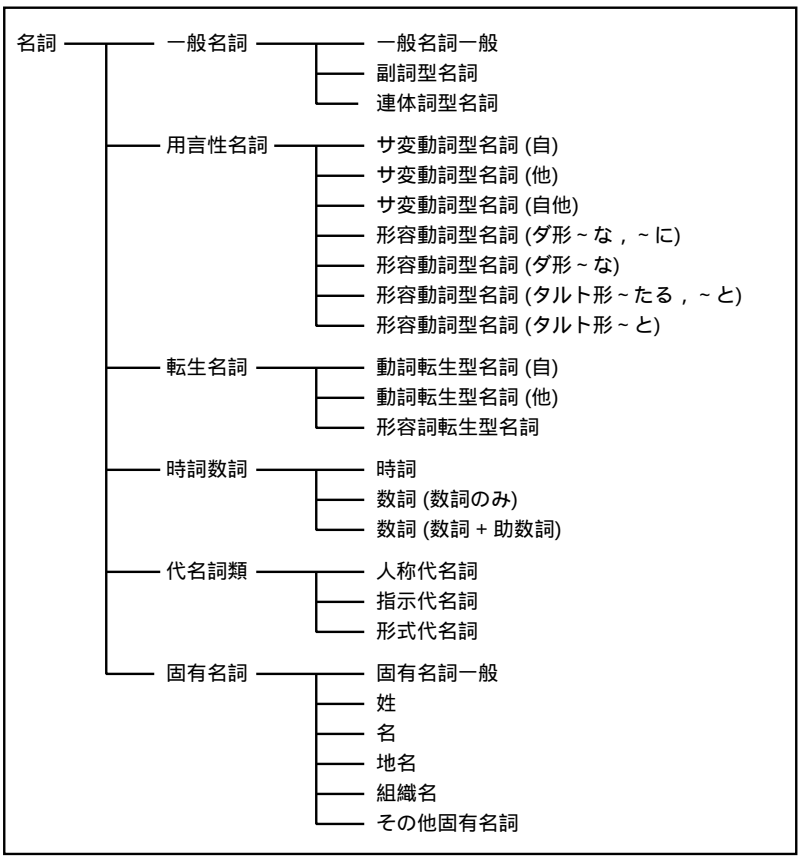

図 10　名詞の文法属性体系

ここで，図 10 の構造を見ると，各文法属性間の段数は少ないから，最終段の文法属性のみを使用した構造規則を生成する．このようなフラットな分類では，決定木を作り，それより式 (1) の形式の規則を作成する方法が便利である．ここでは，決定木生成では，プログラム C4.5(Quinlan 1995) を使用する．

その結果，得られた規則のカバー率は，91.8% ，また，その範囲での正解率は，84.8% であった．従って，全体の正解率は，77.8% である．得られた構造規則のうち，正解率 85% 以上を示したものを表 3 に示す．

表 3　解析精度の良い構造規則 (正解率 85% 以上)

#	文法属性による構造規則	適用される名詞句の例
1	(∗ , <形式名詞> , ∗ : B)	「昔のままの姿」,「大人のための物語」
2	(∗ , ∗ , <形式名詞> : B)	「玄関の石段のところ」,「声楽の勉強のため」
3	(∗ , <指示代名詞> , ∗ : B)	「湿原の向こうの林」,「海の彼方の国々」
4	(∗ , ∗ , <副詞型名詞> : B)	「彼の小説の数々」,「新宿の二丁目の近く」
5	(∗ , ∗ , <指示代名詞> : B)	「窓のガラスの向う」,「逢坂の関の彼方」
6	(∗ , <形容詞転生型> , ∗ : B)	「事件の残酷さの意味」,「もとの静けさのなか」
7	(∗ , <動詞転生型 (自) > , ∗ : B)	「砂のくぼみの中」,「岬のつづきの丘」
8	(<人称代名詞> , <サ変動詞型 (他) > , ∗ : B)	「彼の指揮の下」,「私たちの受験の頃」
9	(<指示代名詞> , ∗ , ∗ : B)	「ここの学校の子」,「こちらの膝の上」
10	(∗ , ∗ , <時詞> : B)	「父の死の直前」,「山本の訓示のあと」
11	(∗ , <サ変動詞型 (自他) > , ∗ : B)	「ピアノの稽古のため」,「司祭の不在の間」
12	(∗ , ∗ , <形容詞転生型> : B)	「漁夫の生活の厳しさ」,「父の声の暗さ」

これらの結果から，以下のことが分かる．

1)　得られた規則のカバー率は，意味属性を使用した場合に比べて若干高い．

2)　得られた規則の正解率は，意味属性を使用した場合に比べて低い．

このうち，1) は，意味属性に比べて文法属性の方がカバー範囲が大きいためと考えられる．また，2) は，意味属性に比べて品詞コードの分類が少ないこと，中でも大半の事例を構成する名詞は「一般名詞」に属すため，分解能が低いことが原因と考えられる．

(2) 文法属性規則の作成

文法属性による規則として，どのような規則があるかについては，従来から検討されており (穴倉，宮崎 1995)，人間による標本分析で比較的容易に推定することができる．ここでは「の型名詞句」において同格の「の」が使用された事例の解析に適用するための「文法属性規則」を考える．

ところで，名詞句「A の B」において，助詞「の」が同格を意味する場合は，以下の二つの場合が代表的である．

(1)　人名を含む同格表現

名詞 A:　意味属性が「人」で，文法属性が「固有名詞 (姓)(名)」でない名詞

名詞 B:　意味属性は「人」で，文法属性が「固有名詞 (姓)(名)」である名詞

(2)　地名を含む同格表現

名詞 A:　意味属性は「地域」で，文法属性が「固有名詞 (地名)」でない名詞

名詞 B:　意味属性は「地域」で，文法属性が「固有名詞 (地名)」である名詞

前節で使用した 1 万件の名詞句のうち，人名，地名を含む同格表現，それぞれ，102 件，13

件に対して，上記の規則を適用した結果によれば，カバー率は，それぞれ，74.5%，100% で，正解率はいずれも 100% であった．

(3)「文法属性規則」と「意味属性規則」の併用

今までの実験結果から見ると，「文法属性規則」に比べて，「意味属性規則」の方が総合的な解析精度は高いと言えるが，表現によっては，「文法属性規則」の方が精度の良い場合もある．そこで，ここでは，両者を組み合わせて使用する場合の効果について評価する．

具体的には，4.1 で得られた「意味属性規則」のそれぞれの正解率と $4.2.1$ で得られた「文法属性規則」の正解率に基づき，以下の手順で解析実験を行う．

1) まず，「文法属性規則」のうち，精度がある一定値 $\eta\%$ 以上の規則を使用して，係り受け解析を行う．
2) 次に，1) で係り受け関係が決定できなかった標本に対して，「意味属性規則」によって係り受け解析を行う．
3) η の値を変えながら，1)，2) の手順を繰り返し，その結果を総合して，最終的な解析精度を評価する．

以上の実験の結果を図 11 に示す．図では，左端の点，右端の点が，それぞれ，「意味属性規則」のみの場合と「文法属性規則」の場合を示しており，中間の 2 点は，$\eta = 85\%$ の場合と $\eta = 70\%$ の場合を示している．

この結果によれば，$\eta = 85\%$ の時，すなわち，2 割の名詞句は，「文法属性規則」，残る 8 割は，「意味属性規則」によって解析されるとき，解析精度は，ほぼ最大で，86.8% となる．

5.3 検討

(1)「逐次型生成」と「同時型生成」の比較

本論文では，「意味属性規則」を生成するに際して，汎用性が高く，数少ない規則でカバー率をあげることを目標に，次元の低い規則から順に生成する方法を考えた．また，各次元の規則の生成では，一度，生成に使用した事例は，事例集合から削除し，残された事例から次の規則を生成する方法 (「逐次型生成」) を採った．しかし，この方法は，事例数が少ないときは，解析精度の上で必ずしも良い方法と言えない可能性がある．すなわち，初めの段階での規則生成では，かなり多くの事例が存在するため，精度の良い規則が生成できるが，規則生成が進むにつれて，残された事例数が減少し，そこから生成される規則の精度が低下することが予想される．この傾向は，事例数 1 万件の場合 (表 2) において，構造規則の精度が，後に生成される規則ほど低下していることからも観察される．

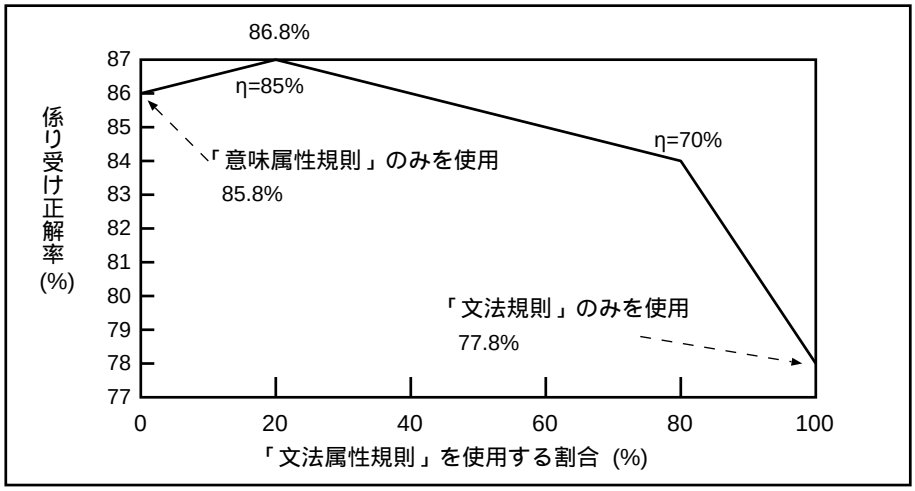

図 11　文法属性による規則と意味属性による規則の併用

そこで，ここでは，規則の生成に使用した事例を捨てないで，各次元の規則を生成する方法 (「同時型生成」と呼ぶ) について実験を行った．ただし，この方法では，一つの事例が異なる次元や異なるタイプの規則の生成で，クラスの異なった規則の生成に使用される可能性があるので，ここでは，得られた構造規則を使用して係り受け解析を行う場合，同次元内の構造規則で適用可能なものはすべて使用することとした．従って，解析では，異なった規則の適用によって異なった係り受け結果が得られる場合が生じる．そこで，係り受け解析においては，以下の方法で係り先を決定した．

＜係り受け解析の手順＞

1)　係り受け解析規則は，一次元規則，二次元規則，三次元規則の順に適用する．

2)　同次元内の複数の規則が適用され異なる係り先が得られた場合は，その次元での判定は保留し，次の次元での結果に従う．

実験の結果を表 4 と表 5 に示す．

表 4　生成された構造規則の数と精度

番号	構造規則の種類	規則のタイプ	得られた規則数		カバー率		精度
1	一次元規則	$(X,*,*,D)$	78.5	小計 223	9.0%	小計 19.3%	91.5%
2		$(*,Y,*,D)$	84.3		8.8%		89.2%
3		$(*,*,Z,D)$	60.3		6.2%		87.6%
4	二次元規則	$(X,Y,*,D)$	974.6	小計 2,859	59.3%	小計 85.8%	90.2%
5		$(*,Y,Z,D)$	937.0		69.8%		91.7%
6		$(X,*,Z,D)$	947.2		67.0%		91.3%
7	三次元規則	(X,Y,Z,D)	2,455		91.0%	91.0%	86.7%
合計		- - - - -	5,528				

表 5　係り受け解析への適用結果

規則の次元	適用事例数	累積頻度	正解率
一次元規則	1,925 (19.3%)	1,925 (19.3%)	91.7%
二次元規則	6,594 (65.9%)	8,519 (85.2%)	90.7%
三次元規則	1,083 (10.8%)	9,602 (96.0%)	68.4%
字面規則	398 (4.0%)	- - - - -	– –
合計	10,000	96.0%	88.4%

これらの結果を「逐次型生成」の場合の結果 (表 1) と比較すると，以下のことが分かる．

1) 「同時型生成法」で生成された規則は，一次元規則，二次元規則に比べて，三次元規則の精度が若干悪いが，「逐次型生成」(表 2) の場合と比べるとかなり向上している．
2) 規則のカバー率は，三次元規則が最大 (91%) で，次元が下がるにつれて，低下する．
3) これらの結果，「逐次型生成」に比べて，「同時型生成」では，規則全体のカバー率が，89.8% から 96% に向上し，解析正解率は，85.8% から 88.4% に向上している．
4) しかし，「同時型生成法」で生成された規則数は，「逐次型生成法」の場合 (1,815 件) に比べてから，約 3 倍 (5,528 件) に増大している．

これより，事例数 1 万件を使用したとき，「同時型生成法」は，「逐次型生成法」に比べて，カバー率が約 6% 向上し，解析精度は，2 ~ 3% 向上することが分かる．しかし，その代わりに生成される規則規則数は，ほぼ 3 倍に増加していることを考えると．事例数の少ないときに使用するのが適切と思われる．

(2) 事例数と構造規則の関係について

表 1 で示されるように，実験では，学習事例に含まれる名詞の種類は，異なり意味属性数から見て，半分弱に止まっており，決して，網羅的とは言えないが，得られた構造規則のカバー率は，表 2 に示されるように 89.8%に上っている．このことから，本方式では，比較的少ない事例から，カバー率の高い構造規則が得られることが分かる．

次に，得られた構造規則の数と精度について見ると，表 2 の結果では，意味属性を使用した構造規則として，名詞句の事例 1 万件から 1,815 件の係り受け構造規則が得られている．本方式では，意味属性によって表現構造とクラスの関係が規定できる事例から構造規則は生成され，それ以外の事例は，字面のままの規則として残されるから，得られた構造規則において，元の事例の持つ情報量は失われない．従って，用例翻訳 (長尾 1984; 佐藤 1992) など，用例そのものを使用する方法に比べ，精度を落とすことなく，言語知識を 1/5 以下に圧縮する効果が期待できる[24] ，また，本方式で得られた構造規則は，意味属性番号や文法属性番号を意味属性名や文

24 得られた規則数 1,815 件には，「字面規則」は含まれていない．これは以下の事情による．すなわち，生成された規則による係り受け解析実験では，「字面規則」が適用された例はないから，この規則を除いても，解析精度は低下しないことである．また，構造規則生成では，事例数が閾値以下の事例からは，構造規則は生成されず，無視されるが，これは，

法属性名に書き換えると，可読性が高いから，人手によってさらに圧縮できる可能性がある．

ところで，最近の記憶装置の価格を考えると，実用上，規則数が多少多いことはあまり問題にならなくなってきた．これに対して，すでに述べたように，言語表現はきわめて多彩であり，構造的，意味的な曖昧性を解消するための知識を人手によって集積するのは，依然として，大変困難な課題である．本論文で提案した方法は，比較的少量の標本から，表現とその解釈に関する精度の良い構造知識を手軽に収集できる方法として，実用性が高いと期待できる．

(3) 規則生成に使用する事例数の閾値について

一般に，精度良い規則を得るには，事例数の多いところから構造規則を生成するのが望ましいが，実験では，一次元規則と二次元規則は，事例数が 2 以上のところから規則を生成し，三次元規則は，事例が 1 つしかない意味属性の組からも構造規則を生成した．これは，三次元規則の生成で，頻度 2 以上の事例から生成した規則より，頻度 1 以上の事例から生成した規則の方が全体として解析精度が良かったためである．

しかし，表 2 では，一次元規則が使用される回数は，1 規則あたり 10 回近くになるのに対して，二次元規則は $4 \sim 5$ 回，三次元規則は 3 回程度と，順に使用回数が減少している．この点から見ると，低次元の規則は．より多くの事例のあるところから生成する方が適切と考えられる．従って，標本数がより多い場合は，事例数の多いところからのみ規則を生成するようにすれば，より精度の良い規則が得られるものと期待される．

6　あとがき

自然言語処理において，さまざまな解釈の曖昧さを解消するための知識を構造規則として記述する方法と，その規則を事例から半自動的に収集する方法を提案した．これは，従来の要素合成法的な方式では解決できない曖昧さの解消を狙ったもので，解釈の曖昧さが問題となる表現を一つの表現単位として扱うことを基本としている．本方式の技術的特徴については，以下の通りである．

本方式では，解釈の曖昧性が問題となる表現を，まず，変数部分と字面の部分からなるパターンで表現した後，構造規則を変数部分に対する制約条件と解釈の組によって定義した．変数部分の記述では，「オールマイティ記号」，「文法属性」，「意味属性」，「字面」の 4 種類の記号の使い分けが可能で，汎用的な規則から個別的で慣用的な表現まで柔軟に表現できる．

次に，生成される規則は，オールマイティ以外の記号が使用される変数部分の数によって次元規則のグループに分類され，各グループの中で汎化が行われる．例えば，N 個の変数を持つ表現パターンの場合，一次元規則から N 次元規則までの規則と字面からなる例外規則を合わせ

用例ベースによる処理の場合と同様で，違いは，あらかじめ事例から規則を生成しておくか，それとも，解析実行時に適用可能な用例をまねるかの差である．

て $N+1$ のグループの構造規則が，順に生成される．汎化は，各次元の特徴空間の中で，木構造で表現された文法属性もしくは意味属性の意味的な包含関係を辿ることにより，容易に実行されるが，このとき，「実際の表現解析では，構造規則は生成された順に適用される」ことを前提に，一度，規則生成に使用された事例を事例集合から削除することにより，汎化領域の拡大と規則数の削減を図っている．

本方式を「A の B の C」の型の名詞句に対する名詞間の係り受け解析規則の生成に適用した結果では，変数部分を意味属性で表現した構造規則の場合，1 万件の学習事例から，一次元規則 198 件，二次元規則 1,480 件，三次元規則 136 件が得られた．そのカバー率は，89.8% であったが，この値は，学習用の標本に含まれる名詞の種類が全体 (約 2,700 種類) の半分以下 (1,000 ~ 1,300 種類) であった点から見てかなり高い．これを使用した係り受け解析では，約 86 %の解析精度が得られた．また，変数部分を文法属性で表した規則と意味属性で表した規則を併用する場合は，解析精度は，1 ~ 2% 向上する．

これらを，2 名詞間の結合強度に還元して評価する従来の方法 (解析精度 72%) と比較すると，3 つの名詞を 1 組として扱うことの重要性が確認できる．また，人間の判断能力と比べると，この種の名詞句では，人間でも係り先の判定に迷うような事例が 10% 近く存在することから，得られた規則の精度は，人間の判断能力にかなり近い値と言える．

なお，提案した方法では，一度，規則生成に使用した事例は学習事例から削除し，残された事例から次の次元の規則を生成する (「逐次生成法」) こととしているが，各次元の規則をすべての事例から生成する方法 (「同時生成法」) では，得られた規則による解析精度は 2 ~ 3% 向上する．しかし，この方法は，事例削除の方法に比べて規則数が 3 倍にも増大する点が問題である．

今後は，提案した方法を複合語解析，数量表現解析など，さまざまな表現解析用の規則生成に適用し，その効果を確認すると共に，より強力な汎化の方法についても検討していきたい．

謝辞

本研究は，NTT コミュニケーション科学基礎研究所，および，文部省の科学研究費補助金の支援を受けて行われたことを記し，関係各位に深謝する．

参考文献

Almuallim, H., Akiba, Y., Yamazaki, T., and Kaneda, S. (1994a). "Induction of Japanese-English Translation Rules from Ambiguous Examples and a Large Semantic Hierarchy." 人工知能学会誌, **9** (5), 730–740.

Almuallim, H., Yamazaki, T., Akiba, Y., Yokoo, A., and Kaneda, S. (1994b). "A Tool for the Acquisition of Japanese-English Machine Translation Rules Using Inductive Leaning Techniques." *The 10th Conference of Artificial Intelligence for Application.*

フセイン・アルモアリム, 秋葉泰弘, 金田重郎 (1997). “木構造を許容する決定木学習.” 人工知能学会誌, **12** (3), 421–429.

Alves, P., de H. W., E., and Furugori, T. (1998). “A Method for Estimating Strength of Association and Its Application to Structural Disambiguation.” *Journal of Natural Language Processing*, **5** (3), 53–65.

穴倉 宮崎正弘 (1995). “構成要素の統語・意味的制約を利用した日本名詞句解析.” 信学技報，NLC94-49, pp. 41–48.

浅野長一郎 江島伸興 (1996). 「基本多変量解析」. 日本規格協会.

麻野間直樹 中岩浩巳 (1999). “目的言語の単語共起情報を利用した訳語選択と未知語の訳出.” 言語処理学会第 5 回年次大会論文集, pp. 442–448.

安西祐一郎 (1989). 「認識と学習」. 岩波書店.

江尻 宮崎正弘 (1998). “名詞間の接続強度と「の」型名詞句の用例を利用した日本語名詞句構造解析法.” 情報処理第 56 回全国大会, 1Q–2.

Haussler, D. (1988). “Quantifing inductive bias: AI leaning algorithm and Variant’s leaning frame-work.” *Artificial Intelligence*, **26** (2), 177–221.

池原悟, 宮崎正弘, 白井諭, 林良彦 (1987). “言語における話者の認識と多段翻訳方式.” 情報処理学会論文誌, **28** (12), 1269–1279.

池原悟, 宮崎正弘, 横尾昭男 (1993). “日英機械翻訳のための意味解析用の知識とその分解能.” 情報処理学会論文誌, **34** (1), 29–43.

Ikehara, S. and Bond, F. (1996). “Approaches to Disambiguation in ALT-J/E.” In *MIDDIM-96, Post-COLING Seminor on Interactive Disambiguation*, pp. 107–117.

池原悟, 宮崎正弘, 白井諭, 横尾昭男, 中岩浩巳, 小倉健太郎, 大山芳史, 林良彦 (1997). 「日本語語彙大系」全 5 巻. 岩波書店.

菊池隆典 白井英俊 (2000). “語彙意味情報に基づく日本語名詞句の意味解析.”. 科学研究費補助金成果報告書，No. 09480067,（研究代表者：郡司隆男），pp. 67-94.

桐澤洋, 池原悟, 村上仁一 (1999). “名詞の訳語選択における意味属性の有効性.” 信学技報，NLC55-5, pp. 29–36.

Kobayashi, Tokunaga, and Tanaka (1996). “名詞間の意味的共起表現を用いた複合名詞の解析.” 自然言語処理, **3** (1), 29–43.

黒橋禎夫 長尾眞 (1994). “並列構造の検出に基づく長い日本語文の構文解析.” 自然言語処理, **1** (1), 35–58.

Mitchell, T. M. (1997). *Machine Leaning*. McGraw Hill.

宮崎正弘, 白井諭, 池原悟 (1995). “言語過程説に基づく日本語品詞の体系化とその効用.” 自然言語処理, **2** (3), 3–25.

Nagao (1988). “A Framework of a Machine Translation between Japanese and English by Analogy Principle.”. in A. Elithorn & Banerji, ed. Artificial and Intelligence, North Holland.

長尾真 (1988).「知識と推論」. 岩波書店.

長尾真編 (1996).「自然言語処理」. 岩波書店.

中井慎司, 伊藤真樹, 池原悟, 白井諭 (1998). “名詞間係り受け解析に必要な単語意味属性の組の最適化.” 情報処理学会第 57 回全国大会, 4R–1.

Nierenburg, S., Carbonell, J., Tomita, M., and Goodman, K. (1992). *Machine Translation; A Knowledge Base Approach*. Morgan Kaufmann Publishers. San Meteo, California.

NTT (1996). *Reference Manual for Morphological Analysis Program ALT-JAWS*.

Pustejovsky, J. (1995). *The Generative Lexicon*. The MIT Press.

Quinlan, J. R. (1993). *C4.5 Program for Machine Leaning*. Morgan Kaufmann. San Mateo, CA.

Quinlan, J. R. (1995). AI によるデータ解析. トッパン.

佐々木美樹 坂本仁 (1995). “文章一括による係り受け関係の解析.” 自然言語処理第 1 回年次大会, pp. 101–104.

佐藤理史 (1992). “事例に基づく翻訳.” 情報処理, **33** (6), 673–681.

島津明, 内藤昭三, 野村浩郷 (1986). “助詞「の」が結ぶ名詞の意味的関係の解析.” 計量国語学, **15** (7), 247–266.

白井諭, 池原悟, 横尾昭男, 木村淳子 (1995). “階層的認識構造に着目した日本語従属節間の係り受け解析の方法とその精度.” 情報処理論文誌, **36** (10), 2353–2361.

Sumita, E. and Iida, H. (1992). “Example-based Transfer of Japanese Adonominal Particles into English.” In *IEICE Trans. Inf. & Syst.*, pp. 585–594.

隅田英一郎, 古瀬蔵, 飯田仁 (1994). “英語前置詞句係り先の用例主導あいまい性解消.” 電子情報通信学会論文誌, **J77-D-2** (3), 557–565.

武田浩一, 浦本直彦, 那須川哲哉, 荻野紫保, 堤泰治郎 (1995). “知識ベースを利用した機械翻訳システム salt2.” コンピュータソフトウエア, **12** (5).

田中穂積 (1989).「自然言語解析の基礎」. 産業図書.

田中穂積監修 (1999).「自然言語処理– 基礎と応用– 」. 電子情報通信学会.

冨浦洋一, 中村貞吾, 日高達 (1995). “名詞句「NP の NP」の意味構造.” 情報処理学会論文誌, **36** (6), 1441–1448.

内海彰, 堀浩一, 大須賀節雄 (1993). “自然言語処理のための形容詞の意味表現.” 人工知能学会誌, **8** (2), 192–200.

宇津呂武仁, 松本裕治, 長尾真 (1993). “二言語対訳コーパスからの動詞の格フレーム獲得.” 情

報処理学会論文誌, **34** (5), 913–924.

Witten, I. H. and Frank, E. (1999). *Data Mining*. Morgan Kaufmann Publishers.

略歴

池原 悟: 1967 年大阪大学基礎工学部電気工学科卒業．1969 年同大学院修士課程終了．同年日本電信電話公社に入社．数式処理，トラフィック理論，自然言語処理の研究に従事．1996 年スタンフォード大学客員教授．現在，鳥取大学工学部教授．工学博士．1982 年情報処理学会論文賞，1993 年同研究賞，1995 年日本科学技術情報センタ賞 (学術賞)，同年人工知能学会論文賞受賞．電子情報通信学会，人工知能学会，言語処理学会，認知言語学会，各会員．

中井 慎司: 1999 年鳥取大学大学院工学研究科知能情報工学専攻博士前期課程修了，現在，インテルコムズ (株) 勤務．在学中は自然言語処理の研究に従事．

村上 仁一: 1986 年，NTT 情報通信処理研究所に入社。音声認識のための自然言語処理の研究に従事。特に trigram の研究をおこなった。1991 年から 1995 年，ATR 自動翻訳電話研究所に出向。1995 年から PB 自動電話番号案内サービスの開発に従事。1997 年に豊橋技術科学大学において論文博士を取得。タイトルは「確率的言語モデルによる自由発話認識に関する研究」。1998 年に鳥取大学工学部知能情報学科に転職，現在に至る。電子情報通信学会および日本音響学会および言語処理学会各会員

(2000 年 5 月 6 日 受付)
(2000 年 7 月 8 日 再受付)
(2000 年 9 月 10 日 採録)