前編集結果を利用した前編集自動化規則の獲得

山口昌也† 乾 伸雄†
小谷善行† 西村恕彦†

機械翻訳システムの解析精度を落とす原因の1つとして，辞書や文法をはじめとした知識の不足があげられる．しかし，すべての知識をあらかじめ網羅的に記述しておくことは困難であり，不足している知識を何らかの方法で獲得できなければならない．さらに，解析誤りや解析不能に直面するのは利用者であることを考えると，不足している知識を利用者から獲得できることが重要である．そこで，本論文では，前編集前後の解析結果から前編集を自動化するための規則の学習手法を提案する．原文をうまく翻訳できるように修正するという前編集は従来から行われており，利用者がシステムに知識を伝達する形態として利用できる．学習される規則は，前編集前後の表層構造間の対応規則として定義され，文法的な知識と語彙的な知識を記述する．学習は漸進的に行われ，(1) 前編集前後の解析結果中に含まれる構成要素の対応関係の認識，(2) 前編集部分の抽出，(3) 既存の規則との合併，といったプロセスを経る．この手法によれば，利用者は前編集を制約する条件を明示的に指定する必要がなく，前編集を行うだけで，システムは前編集を自動化する規則を獲得することができる．

Acquisition of Automatic Pre-edition Rules from Results of Pre-edition

Masaya Yamaguchi,† Nobuo Inui,† Yoshiyuki Kotani† and Hirohiko Nisimura†

In machine translation systems, the incomplete knowledge — for example, a missing lexical item or a grammar rule — is one of the elements that causes the system to fail. It is, however, difficult to describe all the knowledge beforehand. So the system needs to acquire the missing knowledge in a certain way, and it is important to do it from users, considering that it is users who face analysis failures and errors. A traditional way in which users transfer knowledge to the system is by modifying sentences in case of failures so that the system can analyze them well — a process which is called "pre-editing". In this paper, we propose a method of acquiring rules that can automatically pre-edit from the results of user's pre-editing in the previous sessions. Acquired knowledge is defined as rules that make a correspondence between the surface structures before and after a successful pre-editing. The rule can be a grammatical rule or a lexical rule. Our acquiring algorithm acquires the rules gradually, and consists of the following processes: (1) recognizing the elements contained in the result of the syntactic analysis before and after a pre-edition, (2) extracting the pre-edited place, and (3) merging with an existing rule. Our method allows the user to freely specify the conditions that constrain the pre-edition.

1. はじめに

機械翻訳システムに関する研究は，自然言語処理のアプリケーションとして研究が進められ，これまでに多くの成果を得てきた．しかし，現状ではつねに正しい翻訳結果を得られるわけではない[1)]．この原因の1つとして，辞書や文法をはじめとした知識の不備があげられる．知識の質と量はシステムの解析精度を左右する重要な要素であるが，それをあらかじめ網羅的に記述しておくことは困難である．したがって，機械翻訳システムには不足している知識を何らかの形で獲得するための能力が必要である．しかも，その際，解析誤りや解析不能の問題に実際に直面するのはシステム開発者ではなく，利用者であるということから，利用者から獲得できることが重要である．

従来から，機械翻訳システムでは，利用者の知識を解析精度向上に利用するための手段として，前編集を用いてきた．前編集の研究には，入力文の文法を制限するという，制限文法を用いた文書作成支援システム

† 東京農工大学工学研究科
Department of Computer Science, Tokyo University of Agriculture and Technology

に関する研究[2]，前編集場所の特定を行って前編集を効率化する研究[3]がある．これらの研究は，利用者が前編集するのを支援，効率化することに焦点を当てており，利用者の知識をシステムの知識として獲得していない．

また，自然言語処理のための知識獲得は，辞書や文法の知識獲得の研究として，動詞の格フレームの獲得[4]や，文法推論[5]，翻訳規則の獲得[6]に関する研究が盛んに行われている．しかし，これらの学習アルゴリズムに学習用データを与えるためには，学習アルゴリズムが想定するデータに関する知識や言語学的な知識，目標言語に関する知識，といった知識を利用者が持っていなければならない．このことは，利用者にとって負担となる．また，これらの研究では，新たに学習される知識と既存の知識との区別がないため，学習により既存の知識に矛盾が生じることがありうる．

そこで，本論文では，前編集前後の表層構造の組から前編集を自動化する規則の学習方法を提案する．この規則を言い換え規則と呼ぶことにする．この手法では，利用者がシステムに学習用データを伝達する方法として，従来から行われている前編集を使っており，利用者に特別な知識を要求することない．また，3.1節で示すように，言い換え規則は，既存の規則とは独立に学習されるため，既存の規則に矛盾を生じさせることはない．

言い換え規則で，前編集前後の表層構造間の関係を表し，2つの表層構造に含まれる構成要素の対応関係，および，前編集の適用条件を記述する．言い換え規則の持つ，表層構造を変形する規則としての役割は，前編集を自動化するために提案されている書き替え規則[7]やChomskyの変形規則[8]と同一である．しかし，言い換え規則は，前編集前後の解析結果から獲得されるため，学習を前提に形式化されていること，語彙的な知識を記述対象としていること，という点でこれらと異なる．

言い換え規則の学習は，複数の前編集前後の用例から一般的な知識を漸進的に学習していくものである．具体的には，前編集前後の表層構造中の構成要素の対応関係を発見するとともに，言い換え規則の適用条件の合併，一般化を図っていく過程である．ただし，本論文では，複数の用例を合併することに焦点を絞っており，合併した適応条件を一般化することは行っていない．なお，前編集前後の1組の用例は，表層構造は異なるが，意味的に同一な構造を持ったものを想定している．「意味的に同一」であることは，利用者が決定する．また，前編集前の表層構造（翻訳は不能，もしくは，誤った解析結果であるとしても）を得ることが可能であるとし，前編集後の結果はシステムにとって翻訳可能であるとする．

この後，2章で，前編集前後の結果から得られる知識の内容について分類，考察する．3章では，言い換え規則の定義を行う．4章では，言い換え規則を学習するためのアルゴリズムを示す．5章では学習例を示し，提案した学習アルゴリズムの評価を行う．

2. 獲得される知識の分類

すでに述べたように，本研究で想定している前編集前後の文の間には，表層構造は異なるが，意味的には同一，という関係がある．ここでは，この関係を満たす2つの文から得られる知識を，「文法的な知識」，「語彙的な知識」の2つに分類して示すことにする．

2.1 文法的な知識

文法的な知識を得ることができる前編集結果として，次の3つをあげる．これらの前編集から得られる知識は，翻訳可能な表層構造と翻訳不能な表層構造との構造上の対応関係である．文献7)の書き替え規則でも，この(2)，(3)の知識を扱っている．

(1) 構文構造の変形

これは，ある構文を意味的に同一な別の構文に変形するものである．能動態と受動態との間の変形がよく知られた例であるが，次のような例もある．文1で述部にある「多い」が，文1'では「人」を修飾し，文1で「人」を修飾していた「信じた」が述部になる（なお，今後，特に記述がない限り，ダッシュが付いている例文が前編集後の結果であるとする）．この種の前編集結果からは，翻訳可能な表層構造と翻訳不能な表層構造中の構成要素の対応関係を獲得することができる．下の例でいえば，文1の文節が文1'のどの文節に対応するかを推論できる．

文1 その話を信じた人は多い．

文1' 多くの人がその話を信じた．

(2) 縮約展開型[7]

これは，省略されている表層構造上の要素を補完するタイプの前編集である．たとえば，文2で省略されているサ変名詞の語尾のサ変動詞を，文2'では補完している．この前編集結果からは，翻訳可能な表層構造（文2'）から取り除くことが可能な構成要素（文2'の語尾のサ変動詞）を推論することができる．

文2 名前を変えて立候補，落選した．

文2' 名前を変えて立候補し，落選した．

(3) 冗長消去型[7]

翻訳結果に影響しない冗長な表現を削除するもので，

丁寧な表現や持ってまわった表現などを削除することがこれに相当する．下は，後者の例である．このような前編集結果からは，「省略の補完」とは逆に，付加しても同一の意味構造を生成する構成要素（文3の「のである」）を推論することができる．

文3 どうしても行けないのである．

文3' どうしても行けない．

2.2 語彙的な知識

語彙的な知識を獲得可能な前編集結果として，次の2つをあげる．これらからは，表層構造に含まれる構成要素間の語彙的な対応関係を規定する規則を獲得できる．

(1) 語の言い換え

これは，ある語を別の語で置き換えるもので，同義語や類義語での置き換えなどがその例である．文4，4'は「ヘビーだ」から「頑丈だ」への置き換え，文5，5'は「立てる」から「書く」への置き換えである．後者のように，語の置き換えだけにとどまらず，格マーカが変わる場合もある．この前編集から得られる知識は，未知の語彙項目（文4の「ヘビー」）と既知の語彙項目（文4'の「頑丈」）とを結び付ける知識である．

文4 その機械は作りがヘビーだ．

文4' その機械は作りが頑丈だ．

文5 新聞は「全面無罪」と見出しを立てた．

文5' 新聞は「全面無罪」と見出しに書いた．

(2) 句の言い換え

これは，句のレベルで別の表現に置き換えるものである．文6，6'では，慣用表現を通常の表現に置き換えたものである．また，文7，7'は「非力だ」という句を「力がない」と別の表現の句に置き換えている．

文6 彼は飛び出した子供に肝を冷やした．

文6' 彼は飛び出した子供にびっくりした．

文7 その人は非力だね．

文7' その人は力がないね．

3. 言い換え規則

3.1 言い換え規則の概要

言い換え規則では，上記の知識を記述するために，翻訳不能な表層構造から翻訳可能な表層構造への変形規則という形を用いる．この規則は既存の文法規則や辞書の内容とは独立しており，学習する際にそれらを変更することはない．言い換え規則の適用は，システムが表層構造を生成した段階，つまり，構文解析後に行われる．本研究では，構文解析の結果として，入力文の依存構造[9]を想定している．

規則の形式は，白井らの提案している書き替え規則[7]に類似しており，前編集前後の表層構造を組にし，それらに含まれる構成要素間の関係を記述したものとなる．ただし，本研究では，前編集前後の結果から規則を学習していくことを目的としているため，学習に適した規則の形式に拡張した．

ここでは仮に，次のような受動態文から能動態文への前編集を行う言い換え規則を例に考える．言い換え規則は，**図1**，**図2**のように2つの要素からなる．

文a 太郎が先生に学校の教室で誉められる．

文a' 先生が太郎を学校の教室で誉める．

図1が前編集前後の表層構造の組（以後，表層構造組と表記する）で，表層構造は素性構造として表現する．矢印の左側，右側の素性構造は，それぞれ前編集前後の表層構造である．

一方，図2の規則（以後，素性値対応規則と表記する）は，前編集前後の表層構造に含まれる構成要素の対応関係を規定する規則である．この規則の左辺（素性値対応記号と表記する）を図1で素性値として用いることにより，2つの素性構造中の構成要素の対応関係を記述する．それと同時に，とりうる構成要素としての条件となる．たとえば，図1中の V_3 は，左辺の

$$\begin{bmatrix} pred & V_1 \\ voice & られる \\ 活用 & V_2 \\ ガ格 & V_3 \\ ニ格 & V_4 \\ 依存 & V_5 \\ 被依存 & V_6 \end{bmatrix} \rightarrow \begin{bmatrix} pred & V_1 \\ voice & \varepsilon \\ 活用 & V_2 \\ ヲ格 & V_3 \\ ガ格 & V_4 \\ 依存 & V_5 \\ 被依存 & V_6 \end{bmatrix}$$

図1 前編集前後の表層構造の組

Fig. 1 An example pair of feature structures before and after pre-edition.

$$V_1 \rightarrow \begin{bmatrix} 品詞 & 動詞 \\ 活用型 & V_{1a} \\ 語 & V_{1b} \end{bmatrix}$$

$V_{1a} \rightarrow$ 下一段，　$V_{1a} \rightarrow$ 上一段

$V_{1b} \rightarrow *$

$V_2 \rightarrow$ 終止形

$$V_3 \rightarrow \begin{bmatrix} pred & V_{3a} \\ 被依存 & V_{3b} \end{bmatrix}$$

$$V_{3a} \rightarrow \begin{bmatrix} 品詞 & 名詞 \\ 語 & V_{3a'} \\ 意味素性 & V_{3a''} \end{bmatrix}$$

$V_{3a'} \rightarrow *$，　$V_{3a''} \rightarrow *$，　$V_{3b} \rightarrow *$

V_4 ：V_3 と同様

$V_5 \rightarrow \varepsilon$

$$V_6 \rightarrow \begin{bmatrix} デ格 & V_{6a} \end{bmatrix}$$

V_{6a} ：V_3 と同様

図2 素性値対応規則の集合

Fig. 2 Sets of correspondence rules of feature values.

ガ格の格要素と右辺のヲ格の格要素が対応していることを示すと同時に，格要素として，任意の意味素性を持った名詞をとりうると規定している．ただし，素性値'*'，'ε'は，それぞれ任意の素性値，空の素性値を表す．

また，「被依存」素性値は，前編集を行った文節に直接，もしくは，間接的に依存する[9]文節を素性値対応記号で置き換えたものであり，「依存」素性値は，前編集を行った文節が直接，もしくは，間接的に依存する文節を素性値対応記号で置き換えたものである．これについては次章で詳しく説明する．

なお，本論文で想定する表層構造における「活用」素性は，活用語の素性としてではなく，文節の素性として持っている．文節の「活用」素性は，文節最後尾の活用語の「活用」素性であり，個々の活用語は「活用」素性を持たない．したがって，文aの文節「誉められる」の活用形は，終止形となる．

言い換え規則の学習は，2つの素性構造中の構成要素の対応関係を認識して素性値対応規則を生成し，素性値対応規則の右辺を合併，一般化していくことである．

3.2 前編集の妥当性

言い換え規則は，表層構造を変形するための規則であるが，規則中に，変形が起こっている部分だけを含んでいるわけではない．これは，表面的に変形が起こっている部分を記述するだけでは，誤った前編集を行ってしまうからである．たとえば，文2，2'の前編集では，サ変名詞の語尾が補完されているが，つねに補完できるとは限らず，「立候補の届け出を済ませた」のように，純粋に名詞として使われている文の場合は補完してはならない．また，文4，4'での「ヘビーだ」は多義語であり，「頑丈だ」のほかにも「大変だ」(例：その仕事はヘビーだ)という意味もある．したがって，「頑丈だ」の意味で使われている場合だけ，言い換え規則が適用されなければならない．

そこで，本論文では，

類似する表層構造には，同一の前編集を行うことができる

と仮定し，変形部分の周囲の構造を言い換え規則に含めることにより，意味的に同一な別の表層構造へ前編集することを保証する．変形部分以外の周囲の構造は，前編集前後の表層構造において対応関係がとれるため，素性値対応規則として記述される．たとえば，図1では，「被依存」と「依存」素性値がそれにあたる．「被依存」素性値は，文a，a'の「教室で」のように変形が起こらず，文節の単位で対応関係がとれる素性を素性値対応記号（図1のV_6）で置き換えたものである．「依存」素性値は，変形部分が依存する文節を素性値対応記号（図1のV_5）で置き換えたものであり，文a，a'の場合，文末で依存先がないため，εとなっている．

3.3 言い換え規則の定義

図1，2のように，言い換え規則は，表層構造組と，素性値対応規則の集合からなる．素性値対応記号は，それぞれの言い換え規則，つまり，1つの表層構造組ごとに定義される．表層構造組と素性値対応規則の一般的な定義は，次のとおりである．まず，素性構造組は次のように定義される．これは，一般的な素性構造の定義の中に，素性値対応記号を含めたものである．

$$f_L \rightarrow f_R$$

ただし，

$$f_L,\ f_R \subset \boldsymbol{F}$$

$$\boldsymbol{F} = \{\langle a, v\rangle \mid a \in \boldsymbol{A},\ v \in \boldsymbol{V_t} \cup \boldsymbol{V_n} \text{ or } v \subset \boldsymbol{F}\}$$

$\boldsymbol{A}$は素性名の集合，$\boldsymbol{V_t}$ ($\ni \varepsilon$) は（素性構造ではない）素性値の集合，$\boldsymbol{V_n}$は素性値対応記号の集合である．$\boldsymbol{V_t}$の要素であるεは，空の素性値（素性値がないこと）を表す．素性値対応記号は，f_Lとf_Rにおいて1対1対応がとれなければならない．

一方，素性値対応規則は，次のように定義される．素性値対応記号が割り当てられる表層構造中の要素として，語，文節，および，語の素性値がある．詳細は，4.3節で述べる．

$$V \rightarrow f_v$$

ただし，

$$V \in \boldsymbol{V_n},$$

$$f_v \subset \boldsymbol{F},\ \text{もしくは},\ f_v \in \boldsymbol{V_t} \cup \{*\}$$

3.4 言い換え規則の適用

すでに述べたように，言い換え規則の適用は，入力文を構文解析した結果に対して行う．言い換え規則適用の条件は，表層構造組の左辺の素性構造f_Lと入力文の表層構造f_Iとが完全にマッチングしたときで，次のことが成り立つ場合である．

$$\forall i, f_L, f_I \ni \langle a_i, v_i\rangle$$

ただし，$v_i \in \boldsymbol{V_n}$のときは，v_iを左辺に持つ素性値対応規則の右辺で置き換え（今後，「素性値対応記号を展開する」と表記する），再帰的にマッチングするか調べる☆．

言い換え規則の適用結果は，マッチング結果の素性

☆ 言い換え規則には，入力文の表層構造にはない「被依存」，「依存」素性が存在する．したがって，マッチングする場合は，4.3節で述べる方法とは逆の手順で「被依存」，「依存」素性自体を修正し，その修正結果とマッチングする必要がある．

値対応組の右辺である．

4. 言い換え規則の学習

4.1 学習の方針

言い換え規則の学習は，前編集前後の表層構造の組から，対応関係にある構成要素を認識して，その合併，一般化を図っていくことである．本論文では，十分蓋然性のある対応関係として，次の3つを仮定する．実際の例については，この後の学習アルゴリズムの中で示すことにする．

対応関係1： 自立語 w が前編集前後の表層構造中にそれぞれ1つずつ存在する場合，自立語 w の素性構造は対応関係にある．

対応関係2： 対応関係1の条件を満たす活用語の語末に付随する助動詞，助詞に起因する素性（テンス，アスペクト，ムードなど）が，前編集前後の両方の表層構造中に存在し，その値が同一な場合，その素性値は対応関係にある．

対応関係3： 次のいずれかの条件を満たす文節 p が前編集前後の表層構造中にそれぞれ1つずつ存在する場合，それらの素性構造は対応関係にある．

条件1 文節 p に間接・直接的に依存する文節すべての対応関係がとれている．文節 p に依存する文節がない場合も含む．

条件2 文節 p が間接・直接的に依存する文節すべての対応関係がとれている．文節 p が依存する文節がない場合も含む．

4.2 アルゴリズムの概要

本論文で提案する学習アルゴリズムは，前編集前後の表層構造の組を入力として，漸進的に言い換え規則を学習していく．学習アルゴリズムを**図3**に示す．

まず，前編集前後の表層構造の組が入力されると，既存の言い換え規則を適用し，自動的な前編集が可能かチェックする．もし，可能であれば，そのまま終了する．可能でないならば，入力された表層構造の組の構成要素を素性値対応記号で置き換え，新たな言い換え規則として追加する（処理A）．次に，ヒューリスティックスを用いて，前編集による変形を制約する場所を特定し，規則の一般化を図る（処理B）．この後の処理では，新たに生成した言い換え規則と既存の言い換え規則の素性値対応規則との合併を図る．

```
IF 既存の言い換え規則によって前編集部分を変形可能
    THEN 終了
ELSE
    素性値対応規則の生成（処理A）
    前編集部分の抽出（処理B）
    IF 生成された言い換え規則と既存の言い換え規則を合併可能
        THEN 既存の言い換え規則との合併
    ELSE 終了
```

図3 言い換え規則の学習アルゴリズム
Fig. 3 The learning algorithm for paraphrasing rules.

この後の節では，それぞれの処理を詳しく説明する．

4.3 素性値対応規則の生成

既存の言い換え規則で自動的な前編集ができないと分かったら，入力された表層構造中の構成要素の対応関係を認識するため，4.1節で示した対応関係に基づいて，素性値対応規則を生成する．

この後の節で述べるように，対応関係の種類ごとに生成方法があるが，生成は次の順序で行う．それぞれの段階では，条件を満たす対応関係すべてに対して素性値対応規則を生成し，その後，次の段階へ進む．

(1) 対応関係1，2の条件を満たす場合
(2) 対応関係3（条件1）の条件を満たす場合
(3) 対応関係3（条件2）の条件を満たす場合
(4) (2)，(3)で生成した素性値対応記号の右辺に，対応関係1，2の条件を満たすものがあれば，(1)と同様に新たな素性値対応記号で置き換える．

4.3.1 対応関係1，2の条件を満たす場合

4.1節の基準のうち，対応関係1，2の基準を満たす素性値 α は，新たな素性値対応記号 V で置き換え，素性値対応規則 $V \rightarrow \alpha$ を生成する．

たとえば，文aを文a'のように前編集し，**図4**のような表層構造組が学習アルゴリズムに入力されたとする（ただし，ゴシック体になっている素性値の単語は，その単語の素性構造の略記である）．このとき，文aには自立語として，「太郎」，「先生」，「学校」，「教室」，「誉める」があり，これらはすべて文a'に1つずつしか存在しないため，対応関係1の条件を満たす．そこで，それぞれの素性構造を新たな素性値対応記号 $V_1, \ldots, V_5$ で置き換え，それを左辺，置き換えた素性値を右辺に持った素性値対応規則を生成する．

次に，対応関係2の条件を満たす素性値として，「誉める」に付随するテンスの素性値に対して，素性値対応規則（V_6）を生成する．置き換えた結果は，**図5**のようになる．

4.3.2 対応関係3（条件1）の条件を満たす場合

対応関係3（条件1）を満たす文節 p は，変形部分（対応関係がとれない文節）に間接的，もしくは，直接的に依存する文節である．p は素性構造中の1つの素性 $\langle a_p, v_p \rangle$ で表されているため，対応関係1，2とは異なり，素性の単位で新たな素性値対応記号 V へ

$$\begin{bmatrix} pred & \text{誉める} \\ tense & \text{た} \\ voice & \text{られる} \\ \text{ガ格} & \text{太郎} \\ \text{ニ格} & \text{先生} \\ \text{デ格} & \begin{bmatrix} pred & \text{教室} \\ \text{ノ} & \text{学校} \end{bmatrix} \end{bmatrix} \rightarrow \begin{bmatrix} pred & \text{誉める} \\ tense & \text{た} \\ voice & \varepsilon \\ \text{ガ格} & \text{先生} \\ \text{ヲ格} & \text{太郎} \\ \text{デ格} & \begin{bmatrix} pred & \text{教室} \\ \text{ノ} & \text{学校} \end{bmatrix} \end{bmatrix}$$

図4 入力の素性構造組
Fig. 4 Pair of feature structures of input pre-edition.

$$\begin{bmatrix} pred & V_1 \\ tense & V_6 \\ voice & \text{られる} \\ \text{ガ格} & V_2 \\ \text{ニ格} & V_3 \\ \text{デ格} & \begin{bmatrix} pred & V_4 \\ \text{ノ} & V_5 \end{bmatrix} \end{bmatrix} \rightarrow \begin{bmatrix} pred & V_1 \\ tense & V_6 \\ voice & \varepsilon \\ \text{ガ格} & V_3 \\ \text{ヲ格} & V_2 \\ \text{デ格} & \begin{bmatrix} pred & V_4 \\ \text{ノ} & V_5 \end{bmatrix} \end{bmatrix}$$

$V_1 \rightarrow$ ほめる
$V_2 \rightarrow$ 太郎
$V_3 \rightarrow$ 先生
$V_4 \rightarrow$ 教室
$V_5 \rightarrow$ 学校
$V_6 \rightarrow$ た

図5 素性値対応規則の生成例1（対応関係1，2の場合）
Fig. 5 Generation of correspondence rules of feature values (Case of words and feature values consisted in words).

$$\begin{bmatrix} \text{ニ格} & V_3 \\ \text{デ格} & \begin{bmatrix} pred & V_4 \\ \text{ノ} & V_5 \end{bmatrix} \end{bmatrix} \Longrightarrow \begin{bmatrix} \text{ニ格} & V_3 \\ \text{デ格} & \begin{bmatrix} pred & V_4 \\ \text{被依存} & V_5' \end{bmatrix} \end{bmatrix} \quad \left(V_5' \rightarrow \begin{bmatrix} \text{ノ} & V_5 \end{bmatrix} \text{を生成}\right)$$

$$\Longrightarrow \begin{bmatrix} \text{ニ格} & V_3 \\ \text{被依存} & V_7 \end{bmatrix} \left(V_7 \rightarrow \begin{bmatrix} \text{デ格} & \begin{bmatrix} pred & V_4 \\ \text{被依存} & V_5' \end{bmatrix} \end{bmatrix} \text{を生成}\right)$$

図6 素性値対応規則の生成例2（対応関係3，被依存素性値の場合）
Fig. 6 Generation of a correspondence rule of a feature value (Case of phrases modifying to pre-edited phrase).

の置き換えを行う．このとき，対応関係3（条件1）の条件を満たし，かつ，p と同一の依存先を持つ文節が n 個存在する場合は，それらをまとめて素性値対応記号に置き換える．これは，同時に出現する文節として規則化するためである．

対応関係3（条件1）を満たす文節を $p_1,\ldots,p_n$，これらに対応する素性を $\langle a_{p1},v_{p1}\rangle,\ldots,\langle a_{pn},v_{pn}\rangle$ とすると，置き換え前後の素性と生成される素性値対応規則は，次のようになる．このとき，文節単位での対応関係を表すために，特別な素性名「被依存」の素性値として置き換えられる．

置き換え前の素性 $\langle a_p,v_p\rangle$
置き換え後の素性 $\langle$ 被依存$,V\rangle$
生成される素性値対応規則

$$V \rightarrow \begin{bmatrix} a_{p1} & v_{p1} \\ \vdots & \vdots \\ a_{pn} & v_{pn} \end{bmatrix}$$

図5の場合は，「学校の」に依存する文節がないため，対応関係3（条件1）の条件を満たしている．**図6**（上段）のように，V_5' を左辺に持つ素性値対応規則を生成する．この結果，文節「教室で」に依存する文節の対応関係がとれたことになり，対応関係3（条件1）の条件を満たすので，図6（下段）のように，V_7 を左辺に持つ素性値対応規則を生成する．

4.3.3 対応関係3（条件2）の条件を満たす場合

対応関係3（条件2）を満たす文節 p は，変形部分が間接的，もしくは，直接的に依存する文節である．p も対応関係3（条件1）の条件を満たす文節と同様，素性の単位で素性値対応記号 V への置き換えを行う．このとき，p に依存する文節のうち，対応関係3（条件1）の条件を満たしている文節は，p とまとめた形で置き換える．この段階で，対応関係3（条件1）の条件を満たしていない文節 p_t は，変形部分の文節ということになる．p_t が複数ある場合，p が変形部分であると考え，置き換えを行わない．

対応関係3（条件2）を満たす文節 p に対応する素性を $\langle a_{p1},v_{p1}\rangle,\ldots,\langle a_{pn},v_{pn}\rangle$☆，さらに，$p$ に依存する文節の素性を $\langle a_{d1},v_{d1}\rangle,\ldots,\langle a_{dm},v_{dm}\rangle$，$p_t$ に対応

☆ 図4の *pred*, *tense*, *voice* 素性のように複数の素性からなる場合もある．

$$\begin{bmatrix} pred & つける \\ ガ格 & 太郎 \\ ヲ格 & 力 \end{bmatrix} \Longrightarrow \begin{bmatrix} pred & 力 \\ 依存 & V_1 \end{bmatrix} \quad \left(V_1 \rightarrow \begin{bmatrix} pred & つける \\ ガ格 & 太郎 \\ 元素性名 & ヲ格 \end{bmatrix} \text{を生成} \right)$$

図7 素性値対応規則の生成例3（対応関係3，依存素性値の場合）
Fig. 7 Generation of corresponding rule of feature value (Case of phrase modified by pre-edited phrase).

する素性を $\langle a_t, v_t \rangle$ とするとき，置き換え前後の素性構造と生成される素性値対応規則は，次のようになる．生成される素性値対応規則の右辺に「元素性名」素性があるが，これは，p と p_t との関係を保持するための素性である．

置き換え前の素性構造

$$\begin{bmatrix} a_{p1} & v_{p1} \\ \vdots & \vdots \\ a_{pm} & v_{pm} \\ a_{d1} & v_{d1} \\ \vdots & \vdots \\ a_{dn} & v_{dn} \\ a_t & v_t \end{bmatrix}$$

置き換え後の素性構造

$$\begin{bmatrix} 依存 & V \\ pred & v_t \end{bmatrix}$$

生成される素性値対応規則

$$V \rightarrow \begin{bmatrix} a_{p1} & v_{p1} \\ \vdots & \vdots \\ a_{pm} & v_{pm} \\ a_{d1} & v_{d1} \\ \vdots & \vdots \\ a_{dm} & v_{dm} \\ 元素性名 & a_t \end{bmatrix}$$

例として，文bから文b'のように前編集した場合を見てみる．

文b 太郎は英語の力をつける

文b' 太郎は英語の実力をつける

このとき，文節「力を」が依存する文節「つける」には依存先がなく，また，「つける」に依存する文節「太郎は」も対応関係がとれるので，対応関係3（条件2）の条件を満たす．そこで，**図7**のように，依存先の素性とそれに依存する素性をまとめ，素性値対応記号 V_1 を左辺に持つ素性値対応規則を生成する．このとき，「力」は依存先の文節のヲ格だったので，その関係を規則の中に保持するため，「元素性名」素性値とする．

4.4 前編集部分の抽出

前編集前後の文全体をそのまま言い換え規則の表層構造組とすると，一般的な規則を獲得するために多くの入力が必要となってしまう．これは，言い換え規則の妥当性を保証するために，前編集による変形部分だけでなく，非変形部分も言い換え規則に含まれているからである．そこで，学習速度を向上させるために，規則に含まれる非変形部分に対して制限を加える．ここでは，変形部分を制約する非変形部分に関するヒューリスティックとして，非変形部分が前編集の条件となるのは，次の非変形部分の文節であると仮定する．

(1) 変形部分を含む**文節**に直接的に依存する文節

(2) 変形部分を含む**文節**が直接的に依存する文節の主辞

依存文法や動詞の格支配構造辞書（例：IPAL[10]）をはじめとして，現在広く用いられている文法的・語彙的な制約は，直接依存する文節の情報に基づいて記述されているので，この仮定で十分機能すると考えられる．

上記の2種類以外の非変形文節は，変形を制約しないものとして，最大限に一般化を図る．その方法は，次のとおりである．

被依存素性の場合 被依存素性の文節に依存している文節の条件を最大限に一般化する．それには，被依存素性値の素性値対応記号を展開し，その素性構造中に被依存素性値 V ($\in \boldsymbol{V_n}$) が存在したとき，V を左辺に持つ素性値対応規則を削除し，$V \rightarrow *$ を生成する．

依存素性の場合 依存素性の文節が依存している文節の条件を最大限に一般化する．それには，依存素性値の素性値対応記号を展開し，その素性構造中に依存素性値 V ($\in \boldsymbol{V_n}$) が存在したとき，V を左辺に持つ素性値対応規則を削除し，$V \rightarrow *$ を新たに生成する．

図6にこれを適用すると，被依存素性値 V_7 の中に被依存素性値 V_5' があるので，V_5' を左辺に持つ素性値対応規則をすべて削除し，$V_5' \rightarrow *$ を新たに生成する．これにより，任意の文節が文節「教室で」に依存することになる．

4.5 言い換え規則の合併

4.5.1 合併方法

ここでは，新たに生成した言い換え規則と既存の言

い換え規則を合併する．大まかにいえば，既存の言い換え規則の中から，同じ構文構造の表層構造組を持つ言い換え規則を見つけ，2つの言い換え規則の素性値対応規則を合併する．合併の手順は，次のとおりである．

Step1 既存の言い換え規則の中から，同じ構文構造の表層構造組を持つ言い換え規則を見つける．同じ構文構造を持った表層構造を求めるには，素性値対応記号をワイルドカードと考えた状態で，マッチングのとれる表層構造を探せばよい．したがって，同じ構文構造を持った表層構造とは，表層構造 f_a，f_b の差 $D_f(f_a, f_b)$ が次の条件を満たす表層構造であると定義できる．

$\forall i\langle v_{ai}, v_{bi}\rangle \in D_f(f_a, f_b)$

ただし，$v_{ai} \in \boldsymbol{V_n}$，$v_{bi} \in \boldsymbol{V_n} \cup \{\varepsilon\}$，もしくは，

$v_{ai} \in \boldsymbol{V_n} \cup \{\varepsilon\}$，$v_{bi} \in \boldsymbol{V_n}$

なお，v_{ai}，v_{bi} の条件の中に，ε が入っているのは，「被依存」素性値に，任意的に用いることが可能な文節が含まれるのを考慮するためである．たとえば，文 a，a' から生成された言い換え規則と，次の文 a0，a0' から生成される言い換え規則を合併することを許している（文 a0，a0' には，「被依存」素性値が存在しないが，文 a，a' には，「被依存」素性値「学校の教室で」が存在する）．

文 a0 三郎が先生に誉められる．

文 a0' 先生が三郎を誉める．

2つの表層構造の差 $D_f(f_a, f_b)$ の定義は，次のとおりである．ただし，$\boldsymbol{A_a}$，$\boldsymbol{A_b}$ は f_a，f_b に含まれる素性名の集合，v_i は素性名 i の素性値である．

$$D_f(f_a, f_b) = \bigcup_{i \in A_a \cup A_b} D_v(v_{ai}, v_{bi})$$

$$D_v(v_{ai}, v_{bi}) = \begin{cases} \phi & v_{ai} = v_{bi} \\ \langle v_{ai}, v_{bi}\rangle & v_{ai} \neq v_{bi},\ v_{ai},\ v_{bi} \in \boldsymbol{V_n} \cup \boldsymbol{V_t} \\ & \text{ただし，どちらか一方の素性が存在しない場合は，存在しないほうの素性値を } \varepsilon \text{ とする} \\ D_f(v_{ai}, v_{bi}) & v_{ai},\ v_{bi} \text{ がともに素性構造のとき} \end{cases}$$

同じ構文構造の表層構造組となるためには，2つの表層構造組の左辺，右辺それぞれが同じ構文構造を持ち，かつ，表層構造の差が左辺と右辺で同じでなければならない．つまり，新たに生成された表層構造組 $f_{IL} \rightarrow f_{IR}$ と，表層構造組 $f_{EL} \rightarrow f_{ER}$ が同じ構文構造となるためには，次の関係が成り立たなければならない．

$D_f(f_{IL}, f_{EL})$

$= D_f(f_{IR}, f_{ER})$

$\ni \forall i\langle v_{ai}, v_{bi}\rangle$

ただし，$v_{ai} \in \boldsymbol{V_n}$，$v_{bi} \in \boldsymbol{V_n} \cup \{\varepsilon\}$，もしくは，

$v_{ai} \in \boldsymbol{V_n} \cup \{\varepsilon\}$，$v_{bi} \in \boldsymbol{V_n}$

この条件を満たす言い換え規則が見つかった場合は，次の Step2 へ進み，見つからなかった場合，学習アルゴリズムは終了する．

Step2 $D_f(f_a, f_b)$ の要素として，$\langle v_{ai}, v_{bi}\rangle$ が存在するとき，素性値対応規則 $v_{ai} \rightarrow f_{ai}$ を削除し，新たに $v_{bi} \rightarrow f_{ai}$ を生成する．ただし，$v_{bi} = \varepsilon$ の場合は，f_b 中の v_{bi} に対応する素性値を v_{ai} に置き換え，$v_{ai} \rightarrow \varepsilon$ を追加する．以上のことを $D_f(f_a, f_b)$ のすべての要素に対して行う．

Step3 Step2 で合併を行った場合は，Step1 に戻り，再び，合併結果の言い換え規則と既存の言い換え規則の合併を試みる．

4.5.2 合 併 例

例として，図4の言い換え規則がすでに獲得されている段階で，次のような前編集が行われたとする．

文 c 次郎が先生に厳しく叱られる．

文 c' 先生は次郎を厳しく叱る．

このとき，新たに生成された言い換え規則の表層構造組を $f_{cL} \rightarrow f_{cR}$，文 a，文 a' から生成された言い換え規則の表層構造組を $f_{aL} \rightarrow f_{aR}$ としたとき，

$D_f(f_{cL}, f_{aL}) = \{\langle$ 次郎, 太郎 $\rangle$,

$\langle$[厳しく], [教室で]$\rangle$,

$\langle$ 叱る, 誉める $\rangle\}$

となり，次の素性値対応規則が新たに追加される．ただし，本来，差の要素は素性値対応記号であるが，見やすさのために，具体的な素性値に展開して示している．また，角括弧は括弧内の文節の素性構造を表す．

$V_1 \rightarrow$ 叱る, $V_2 \rightarrow$ 次郎, $V_7 \rightarrow$ [厳しく]

4.6 非変形部分の合併

4.6.1 合 併 方 法

非変形部分の合併，つまり，「被依存」素性値，または，「依存」素性値を合併する場合は，素性値対応記号を含んだ素性値を合併する可能性があるので，合併した素性値どうしでさらなる合併が可能かもしれない．そこで，次の手順でさらなる合併を行う．

Step1 新たに生成した素性値対応規則 $V \rightarrow f_{new}$ と既存の素性値対応規則 $V \rightarrow f_{e1}, \ldots, V \rightarrow f_{en}$ としたとき，f_{new} と $f_{e1}, \ldots, f_{en}$ それぞれの合併を試みる．

Step2 もし，f_{new} と f_{ei} が合併可能で，合併した結果 f'_{ei} となったとき，$V \rightarrow f_{ei}$ と $V \rightarrow f_{new}$ を削除し，$V \rightarrow f'_{ei}$ を追加する．

Step3 Step2で合併を行った場合は，Step1に戻り，再び，追加した素性値対応規則と既存の素性値対応規則との合併を試みる．

4.6.2 合 併 例

合併の例として，文b，文b'の前編集結果のかわりに，次のような前編集結果が入力された場合を考える．

文d 太郎が先生に学校の図書館でやさしく誉められる．

文d' 先生は太郎を学校の図書館でやさしく誉める．

この前編集結果から生成される表層構造組 $f_{dL} \rightarrow f_{dR}$ と文a，文a'から生成される表層構造組 $f_{aL} \rightarrow f_{aR}$ との差 $D_f(f_{dL}, f_{aL})$ は，次のようになり（ただし，本来，「やさしく」，「図書館」，「教室」は，素性値対応記号で置き換えられているが，見やすさのために展開してある），

$$D_f(f_{dL}, f_{aL}) = \{\langle v_{dL1}, v_{aL1} \rangle\}$$

$$v_{dL1} \rightarrow \left[\begin{array}{ll} \text{連用修飾} & \text{やさしく} \\ \text{デ格} & \textbf{図書館} \end{array}\right]$$

$$v_{aL1} \rightarrow \left[\begin{array}{ll} \text{デ格} & \textbf{教室} \end{array}\right]$$

v_{dL1} と v_{aL1} は被依存素性値なので，さらに合併を行うため，$D_f(f_{dL1}, f_{aL1})$ を求める（ただし，$v_{dL1} \rightarrow f_{dL1}$，$v_{aL1} \rightarrow f_{aL1}$ とする）．その結果は，次のとおりである（ここでも，見やすさのため，素性値対応記号は展開してある）．

$$D_f(f_{dL1}, f_{aL1}) = \{\langle \text{やさしく}, \varepsilon \rangle, \langle \textbf{図書館}, \textbf{教室} \rangle\}$$

この結果，修正，および，新たに生成される素性値対応規則は，次のとおりである．

$$V_5 \rightarrow \left[\begin{array}{ll} \text{デ格} & \left[\begin{array}{ll} pred & V_{5a} \\ \text{被依存} & V'_6 \end{array}\right] \\ \text{連用修飾} & \left[\begin{array}{ll} pred & V_{5b} \end{array}\right] \end{array}\right]$$

$V_{5a} \rightarrow$ **図書館**

$V_{5b} \rightarrow$ [やさしく]

$V_{5b} \rightarrow \varepsilon$

$V'_6 \rightarrow *$

5. 学習実験，および，評価

5.1 学 習 例

ここでは，提案した学習アルゴリズムを使った学習例を2つ示す．なお，学習システムへの入力となる構文解析結果は，人手で作成した．

学習例1 文法的な知識の学習例として，受動態（助動詞「られる」による）から能動態文への前編集を行い，前編集前後の結果をシステムへの入力とした．用例に利用した文は50文であり，用例の抽出は次のように行った．まず，毎日新聞91年度版[11]の1月分（総文数61619文）から，助動詞「られる」を含む文（2103文）を抽出し，その中から，受動態から能動態への前編集が可能な文（654文）を選別し，さらに，その中から無作為に50文抽出した．前編集の一例として，次の前編集をあげる．

文e 末増容疑者はほかのホテル従業員**に取り押さえられた**．

文e' 末増容疑者**を**ほかのホテル従業員**が取り押さえた**．

学習例2 語彙的な知識の学習例として，「カラー」から「特色」への前編集を行い，前編集前後の結果をシステムへの入力とした．用例に利用した文は50文であり，用例の抽出は次のように行った．まず，毎日新聞91，92，93年度版[11)~13)]（総文数2415537文）から，名詞「カラー」（複合名詞も含む）を含む文（1830文）を抽出し，その中から，「カラー」から「特色」への前編集が可能な文（130文）を選別し，さらに，その中から無作為に50文抽出した．前編集の一例を次に示す．

文f 積極経済運営の宮沢氏の**カラー**を守る方針だ．

文f' 積極経済運営の宮沢氏の**特色**を守る方針だ．

学習例1，学習例2の学習結果として，表層構造組，および，素性値対応規則の生成数を**表1**にまとめる．なお，素性値対応規則の項目の内容は，それぞれ対応関係1，2，3に基づいて生成された素性値対応規則の数である．括弧内の数字は，合併した素性値対応規則の数の平均値（同一の素性値対応記号を左辺に持つ素性値対応規則が平均でいくつ存在するか）を表す．また，学習例1，学習例2で得られた言い換え規則の一部を，それぞれ**図8**，**図9**に示す．ただし，これらの図のうち，素性値対応規則に含まれている被依存素性

表1 生成された言い換え規則の内容
Table 1 Contents of generated paraphrasing rules.

	学習用例数	表層構造組	素性値対応規則
学習例1	50	11	117/89/25 (1.84/1.79/3.24)
学習例2	50	3	27/20/6 (2.81/1.05/4.83)

$$\begin{bmatrix} pred & V_{a1} \\ ガ格 & V_{a2} \\ 被依存 & V_{a3} \\ 依存 & V_{a4} \\ voice & られる \end{bmatrix} \rightarrow \begin{bmatrix} pred & V_{a1} \\ ヲ格 & V_{a2} \\ 被依存 & V_{a3} \\ 依存 & V_{a4} \\ voice & \varepsilon \end{bmatrix}$$

$V_{a1} \rightarrow$ 加える | 考える | 設ける

$V_{a2} \rightarrow$ 協議 | 修正 | 方法

$$V_{a3} \rightarrow \begin{bmatrix} ニハ & V_{a31} \\ デ格 & V_{a32} \end{bmatrix}$$
$$\rightarrow \varepsilon$$

$V_{a31} \rightarrow$ 同法案

$V_{a32} \rightarrow$ 要求

$V_{a4} \rightarrow \varepsilon$

$$\rightarrow \begin{bmatrix} pred & V_{a41} \\ 元素性名 & 連用修飾 \end{bmatrix}$$

$V_{a41} \rightarrow$ プレゼントする

$$\begin{bmatrix} pred & V_{b1} \\ ニ格 & V_{b2} \\ 被依存 & V_{b3} \\ 依存 & V_{b4} \\ voice & られる \end{bmatrix} \rightarrow \begin{bmatrix} pred & V_{b1} \\ ガ格 & V_{b2} \\ 被依存 & V_{b3} \\ 依存 & V_{b4} \\ voice & \varepsilon \end{bmatrix}$$

$V_{b1} \rightarrow$ 尋ねる | 鍛える

$V_{b2} \rightarrow$ 新聞記者 | 伊予田良一

$$V_{b3} \rightarrow \begin{bmatrix} 連用修飾 & V_{b31} \end{bmatrix}$$
$$\rightarrow \varepsilon$$

$V_{b31} \rightarrow$ 後（例：筋肉を鍛えた後，試合に出た）

$V_{b4} \rightarrow \varepsilon$

$$\rightarrow \begin{bmatrix} pred & V_{b41} \\ 元素性名 & 連用修飾 \end{bmatrix}$$

$V_{b41} \rightarrow$ 照れる | つける

図 8 学習例 1

Fig. 8 Examples of acquired rules (1).

$$\begin{bmatrix} pred & カラー \\ 被依存 & V_{c1} \\ 依存 & V_{c2} \end{bmatrix} \rightarrow \begin{bmatrix} pred & 特色 \\ 被依存 & V_{c1} \\ 依存 & V_{c2} \end{bmatrix}$$

$$V_{c1} \rightarrow \begin{bmatrix} 連体修飾 & V_{c11} \end{bmatrix}$$

$V_{c11} \rightarrow$ 豪快な

$$\rightarrow \begin{bmatrix} ノ & V_{c12} \end{bmatrix}$$

$V_{c12} \rightarrow$ 独自 | 本来 | 新聞

$\rightarrow \varepsilon$

$$V_{c2} \rightarrow \begin{bmatrix} pred & V_{c21} \\ 元素性名 & ガ格 \end{bmatrix}$$

$V_{c21} \rightarrow$ 打ち出せる | 出る

$$\rightarrow \begin{bmatrix} pred & V_{c22} \\ 元素性名 & ヲ格 \end{bmatrix}$$

$V_{c22} \rightarrow$ 出す | 持つ | 打ち出す

$$\begin{bmatrix} pred & \begin{bmatrix} 語幹 & カラー \\ 語尾 & \varepsilon \end{bmatrix} \\ 被依存 & V_{d1} \\ 依存 & V_{d2} \end{bmatrix}$$

$$\rightarrow \begin{bmatrix} pred & \begin{bmatrix} 語幹 & 特色 \\ 語尾 & \varepsilon \end{bmatrix} \\ 被依存 & V_{d1} \\ 依存 & V_{d2} \end{bmatrix}$$

$$V_{d1} \rightarrow \begin{bmatrix} ノ & V_{d11} \\ ノガ & V_{d12} \end{bmatrix}$$

$V_{d11} \rightarrow$ 野球

$V_{d12} \rightarrow$ 攻める

$V_{d2} \rightarrow \varepsilon$

図 9 学習例 2

Fig. 9 Examples of acquired rules (2).

は省略し，単語の素性構造は，ゴシック体で略記した．

5.2 学習した規則の適用例

ここでは，学習した規則の妥当性を検証するため，学習した規則を closed データと open データ，さらに，負例データ（言い換え規則を適用してはいけない例．たとえば，「カラー」の場合，「色」を意味する「カラー」には言い換え規則を適用してはいけない）に対し適用する．

closed データは，学習に使った用例の前編集前の表層構造である．open データは，前節と同様の方法で，それぞれ 50 文ずつ無作為に抽出した．ただし，open データと closed データに重複するデータはない．負例データは，学習例 1，学習例 2 で抽出した助動詞「られる」を含む文（2103 文），名詞「カラー」（複合名詞も含む）を含む文（1830 文）の中から前編集してはいけない文をそれぞれ 50 文ずつ無作為に抽出した．

適用結果を**表 2**（一般化なしの行）に示す．“close”，“open”の項目の内容は，左から，言い換え規則が適用されるデータ数/正解を含むデータ数/解が正しく一意に決定するデータ数，を表す．“負例”の項目は，言い換え規則を適用しなかったデータ数を示す．

表 2 言い換え規則の適用結果

Table 2 Results of application of paraphrasing rules.

	closed	open	負例
学習例 1			
一般化なし	50/50/50	0/0/0	50
一般化あり	50/50/46	19/17/16	40
学習例 2			
一般化なし	50/50/50	6/6/6	50
一般化あり	50/50/50	37/37/37	50

1 章で述べたように，本研究の焦点は，言い換え規則の合併にあり，合併した素性値対応規則の一般化は行っていない．ただし，この状態だと，closed データに適用できる規則とはなりにくく，本研究で行った合併と一般化の妥当性を検証することが困難である．そこで，ここでは，合併した素性値対応規則の一般化を行ったときの適用精度の見込みを得るため，対応関係 1，2 で生成された規則の一般化を行い，上と同様の実験を行った．具体的には，対応関係 1 の「語」素性と対応関係 2 の素性を最大限に一般化した（右辺に，素性値 * を持つ規則を作った）．適用した結果を表 2

(一般化ありの行)に示す.

5.3 評 価

5.3.1 学習結果の性質

まず,学習された規則の性質について見てみる.図8,図9にその一部を示したが,言い換え規則は,

学習例1の場合 生じている格の変形の組合せごと

学習例2の場合 「カラー」を名詞,複合名詞の一部として使う用法,文の末尾で体言止め(例:「攻めるのがうちの野球のカラー.」)で使う用法

というように生成された.このように,生成される言い換え規則(表層構造組)の数は,前編集対象に左右され,受動態から能動態への変形のように,変形の組合せが多いと,生成される表層構造組の数も多くなる.その結果,複数の言い換え規則に分散して合併が起こるため,素性値対応規則の合併数が少なくなる.実際に,表1でも,生成される言い換え規則の数が少ない学習例2のほうが,対応関係1,3に基づいて生成された素性値対応規則の合併数が少ないのが分かる.なお,対応関係2に基づいて生成された素性値対応規則の合併数が少ないのは,変形部分の「カラー」を連体修飾する述語が少なかった(50文中7文)ためである.

5.3.2 学習結果の妥当性

次に,学習された規則の妥当性について見てみる.表2(一般化なし)を見ると,closedデータに対しては,2つの学習例ともに,規則の誤適用☆,過適用☆☆なく,正しく規則を適用できている.しかし,openデータに対しては,学習例1は0%,学習例2では12%(解が正しく一意に決定するデータ数の割合)であった.これは,素性値対応規則を合併した際に,その右辺を一般化していないことが一因である.これについては,この後述べる.学習例2のほうが,適用できるデータの数が多かったのは,構文的な言い換えの学習例1よりも語彙的な言い換えの学習例2のほうが,一定の文脈で使われるためだと思われる.最後に,負例に対する規則の適用だが,規則が適用されることはなかった.

一方,一般化を行った表2(一般化あり)を見ると,openデータに対する適用がうまくいくようになり,解を正しく一意に決定するデータ数の割合は,学習例1が32%,学習例2が74%であった.学習例2では,openデータ,closedデータともに,規則の過適用,誤適用はなく,また,負例への規則の適用が起こることはなかった.しかし,学習例1では,規則の過適用/誤適用がclosedデータに対して50文中4文/50文中0文,openデータに対して17文中1文/19文中2文,負例データに対する規則の適用が50文中10文あった.

対応関係1の「語」素性と対応関係2の素性の一般化の程度を最大限にするという条件のもとで,規則の過適用,誤適用が多くなかったことは,今回学習した規則の妥当性を示すものである.

今後の課題としては,次のようなものがある.

- 今回は,合併した素性値対応規則の一般化の程度を最大限にしたが,動詞の格支配構造の獲得の研究[14),15)]で行われているように,うまく一般化の程度を制御するしくみを考える必要がある.
- 学習例1の結果にあるとおり,負例データに対する規則の適用が全体の20%と比較的大きい.したがって,佐藤ら[6)]が行っているような負例からの規則の獲得が必要である.
- 学習例1のような文法的な知識では,格助詞などの助詞の役割は大きい.したがって,openデータに対する規則の適用を多くするためには,今回行った,合併した素性値対応規則の一般化のほかに,対応関係3の一般化,つまり,格助詞をはじめとして,文節と文節との関係を決定する助詞に関する一般化も考慮する必要がある.

6. おわりに

本論文では,機械翻訳システムにおいて不足している知識を利用者から獲得することを目指し,前編集前後の結果から前編集を自動化するための規則,「言い換え規則」の学習方法を提案した.本手法を用いることにより,システムに規則を伝達しているということを利用者に意識させずに,規則を獲得することができる.特に,利用者が前編集を制約する条件を明示的に示す必要がない.提案した学習アルゴリズムによって得られた言い換え規則をopenデータ,closedデータ,負例データに適用した結果,妥当性のある規則を学習できることが分かった.

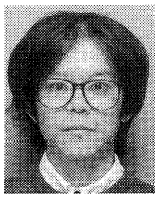
今後の課題として,合併した素性値対応規則,および,文節と文節との関係を決定する助詞に関する一般化,負例からの規則の獲得が必要であることが分かった.本手法で学習された結果は,自動的な前編集だけでなく,前編集場所と前編集候補を利用者に提示することにも利用できる.これをインタエディットに応用すれば,複数の前編集結果を利用者が選択することができ,また,その選択結果は,合併した素性値の一般化を制御する際の重要な情報になると考えている.

☆ 正解を含まず,誤った適用しかしない場合
☆☆ 正解は含むが,誤った適用も行う場合

参考文献

1) 長尾 眞：新しい機械翻訳のための自然言語処理，人工知能学会誌，Vol.11, No.4, pp.500–506 (1996).
2) 長尾 眞，田中伸佳，辻井潤一：制限文法にもとづく文章作成援助システム，情報処理学会自然言語処理研究会資料 44–5，情報処理学会 (1984).
3) 平井章博，梶 博行，芹沢 実：機械翻訳向け前編集のための日本語係り受け構造の曖昧性検出方式，情報処理学会論文誌，Vol.31, No.10, pp.1425–1437 (1990).
4) 宇津呂武仁，松本裕治，長尾 眞：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，Vol.34, No.5, pp.913–924 (1993).
5) 森 信介，長尾 眞：タグ付きコーパスからの統語規則の獲得，情報処理学会論文誌，Vol.37, No.9, pp.1688–1696 (1996).
6) 佐藤理史，長尾 眞：文法推論に基づいた翻訳文法の学習法式，コンピュータソフトウェア，Vol.4, No.4, pp.56–68 (1986).
7) 白井 諭，池原 悟，川岡 司，中村行宏：日英機械翻訳における原文自動書き替え型翻訳方式とその効果，情報処理学会論文誌，Vol.36, No.1, pp.12–21 (1995).
8) 郡司隆男：自然言語の文法理論，産業図書 (1987).
9) 長尾 眞（編）：自然言語処理，第4章「構文解析」，岩波書店 (1996).
10) 情報処理振興事業協会技術センター：計算機用日本語基本動詞辞書 IPAL (Basic Verbs)—辞書編 (1987).
11) 毎日新聞社：CD–毎日新聞'91 データ集 (1991).
12) 毎日新聞社：CD–毎日新聞'92 データ集 (1992).
13) 毎日新聞社：CD–毎日新聞'93 データ集 (1993).
14) Takiguchi, N., Xie, J. and Kotani, Y.: Acquisition of Semantic Feature on Case Frame Structure, *Proc. NLPRS '91*, pp.337–344 (1991).
15) 春野雅彦：最小汎化を用いたコーパスからの動詞格フレームの学習，「自然言語処理における学習」シンポジウム論文集，pp.9–16 (1994).

(平成9年 2 月4日受付)
(平成9年11月5日採録)



山口 昌也（学生会員）

1968年生．1994年東京農工大学工学研究科博士前期課程修了．現在，同大学院博士後期課程に在学中．自然言語処理の研究に従事．言語処理学会会員．

乾 伸雄（正会員）

1963年生．1987年東京農工大学大学院工学研究科修了．1991年より同大学工学部助手．人工知能，自然言語処理の研究に従事．

小谷 善行（正会員）

1949年生．1971年東京大学工学部計数工学科卒業．1977年同大学院博士課程修了，同年より東京農工大学に勤務．現在同大学工学部教授（電子情報工学科情報工学講座）．人工知能，知識処理，自然言語処理，知識獲得，ゲームシステム，ソフトウェア工学，教育工学の研究に従事．電子情報通信学会，人工知能学会等会員，コンピュータ将棋協会会長．最近では，多量データからの知識獲得に興味を持っている．

西村 恕彦（正会員）

1935年生．現在，東京農工大学工学部教授．工学博士．プログラミング言語，暗号解析，自動翻訳の研究に従事．