

機械翻訳のための日英文型パターン記述言語

池原 悟*¹ 宮崎 正弘*² 佐良木 昌*³ 池田 尚志*⁴
白井 諭*⁵ 村上 仁一*¹ 徳久 雅人*¹

*¹ 鳥取大学工学部知能情報工学科 {ikehara,murakami,tokuhisa}@ike.tottori-u.ac.jp

*² 新潟大学工学部情報工学科 miyazaki@ie.niigata-u.ac.jp

*³ 長崎純心大学 saraki@st.rim.or.jp

*⁴ 岐阜大学工学部応用情報学科 ikeda@info.gifu-u.ac.jp

*⁵ NTTアドバンステクノロジー株式会社 shirai@nlp.ntt-at.co.jp

最近、等価的類推思考の原理による機械翻訳方式が提案されている。この方式は、原言語と目的言語の表現の意味的な関係を意味類型パターンとして記述した「意味類型知識ベース」を必要とするが、これを構築するには、多種多様な対訳例文の表現から、相互独立で汎用性の高い文型パターンを如何に抽出して記述するか重要となる。そこで、本稿では、日英両言語の対訳例文から文型パターンを抽出する方法について論じ、両文型パターンを記述するための文型パターン記述言語を提案する。また、収集された文型パターンと翻訳用の入力文を照合する際の基準についても述べる。

キーワード：文型パターン、機械翻訳、パターン記述言語、汎化、意味的等価変換

Sentence Pattern Description Language for Japanese to English MT

Satoru Ikehara*¹ Masahiro Miyazaki*² Masashi Saraki Naoshi Ikeda
Satoshi Shirai*⁵ Jin'ichi Murakami Masato Tokuhisa

*¹ Faculty of Engineering, Tottori University

*² Faculty of Engineering, Niigata University

*³ Nagasaki Junshin Catholic University

*⁴ Faculty of Engineering, Gifu University

*⁵ NTT Advanced Technology Corporation

Recently a new machine translation method based on equivalent analogy has been proposed. In order to realize this method, it is required to build "Semantically Typological Knowledge-base for Languages" which has the semantic relations between an original language and a target language. In this knowledge-base, one of the most important issues will be how to attain mutual independency and high cover rate for sentence patterns obtained from huge amount of bi-lingual corpus. This paper discusses the way to define the sentence patterns and proposes specifications for sentence pattern description language.

Key Words: Sentence Pattern, Machine Translation, Pattern Description Language, Generalization, Semantically Equivalent Transfer Method

1. はじめに

要素合成法を基本とした従来の機械翻訳方式の限界を超えるには、文構造とその意味を一体的に扱う仕組みを実現することが重要である。このような方法の実現を試みたものとして多段翻訳方式[池原87]の研究がある。この方式は、原言語から意味のまとまる表現構造を取り出し、それを「構造的な意味の単位」として目的言語に変換することを目指している。「構造的な意味の単位」としては、用言、および、それと共に起する名詞の關係に着目して、これを結合価パターンの形式で記述している[池原97]。また、適用範囲の拡大等をねらって、入力文の表現を、話者の対象の概念化された認識を表す「客体的表現」と話者自身の概念化されない感情と意志を表す「主体的表現」[三浦67]とに分離し、前者の中から「構造的な意味の単位」を抽出し翻訳規則を適用する仕組みを実現している。

この方法により、単文レベルでの訳文品質（特に用言の訳語選択）は従来に比べて格段に向上した[金出01]が、まだ2つの問題が残されている[池原01]。その第1は、意味の単位と見なされる原言語の表現構造に対して、単一の目的言語表現が対応づけられるため、文脈上、不適切な表現への翻訳が防止できないことである。第2は、複文（埋め込み節を持つ文）や重文（接続構造を持つ文）では、表現全体に対する意味の非線形性を扱う仕組みのないことである [Ikehara 01]。

これに対して、最近、「等価的類推思考の原理による機械翻訳方式」[池原02]が提案された。この方式は、「意味類

型論」の立場から、原言語と目的言語の表現から非線形な表現構造を取り出して文型パターン化した後、意味的同等性に着目して類型化すること、また、意味的に類型化された文型パターン間を「類推思考の原理」[市川63]によって対応づけることの2つの仕組みから構成される。言語表現の意味類型化[有田87]は、「類推思考の原理」による言語間の写像を可能とするためのものである。なお、線形要素は、従来の要素合成法を基本とする方法で変換され目的言語の生成段階で結合される。

この方式の第1の特徴は、異なる言語間の文型パターンを「論理的意味範疇」によって意味的に対応づけられる点である。原言語と目的言語の表現の対応關係は、従来のパターン翻訳のように1対1ではなく、一般に、複数対複数の關係になる。これは、原言語側から見れば、1対複数の關係で目的言語表現に対応づけられることであり、第1の問題の解決が期待される。また、第2の特徴は、「多段翻訳方式」で使用された結合価パターンの適用対象が単文構造に限定されていたのに対して、任意の言語表現を対象としていることである。現在、重文（接続のある文）、複文（埋め込みのある文）および複重文（埋め込みと接続のある文）を対象に研究試作を行っていて、第2の問題の解決も期待される。

ところで、この方式を実現するには、原言語、目的言語の表現の意味類型パターンを収録した「意味類型知識ベース」を構築することが必要であり、収録された意味類型パターンの網羅性が重要である。

現在、大量の日英対訳コーパスを対象に日本語表現と英語表現の非線形部分抽出して文型パターン化し、得られた文型パターンを縮退させたのち意味類型化することによって「意味類型知識ベース」構築することを計画している。品質の良い機械翻訳を実現するためには、日英対訳コーパスから汎用的で実用的な文型パターンを作成する必要がある。本稿では日英の文型パターンを記述する方法と記述言語について検討する。

2. 文型パターン化の方法

文型パターンの作成においては、(1)文型パターン相互間の無矛盾性を保ちながら、(2)いかに被覆率を向上させるかが最大の問題である。本章では、相反するこれらの事象を工学的に調和させる方法について述べる。

2.1 文型パターン化の原則とレベル構成

(1) 文型パターン化の原則

収集した日英対訳文には、強度に文脈に依存した場合など文単位の対訳関係としてパターン化することは適切と言えないものが含まれる。文型パターン化では単独の日本語から対応する英語文が得られるような対訳用例を使用する。

(2) 汎化のレベル構成

訳文の品質を向上させるためには、より個別で具体的な文型パターンを作成することが望まれるが、被覆率を向上させるためには、なるべく汎用的な文型パターンが望まれる。そこで、個性性と汎用性を同時に実現するため、対訳コーパスから順に表1に示す3レベルの文型パターンを作成する。表1で、「変数化」とは、標本文中の線形要素（他の要素に置き換えても表現構造の変化しない部分）を記号（変数）に置き換えることである。

表1 汎化のレベルとその目標

汎化レベル	汎化の内容
単語レベル	文中の単語の内、線形要素と見なせる自立語を変数化する。また、文型上不要な非線形要素を任意化すると共に、字面要素について、可能な範囲でグループ化する。
句レベル	文全体から見て線形要素と見なせる名詞句、動詞句を変数化すると共に、単語レベルで作成された文型パターンの単語変数を可能な限り句の変数に置き換える。
節レベル	文全体から見て線形要素と見なせる節を変数化する。

各レベルのパターン化において複数の同一パターンが得られたときは、それを一つに縮退させる。また、着目している低位のレベルの文型パターンから高位のレベルの文型パターンが作成できない場合は、汎化は元のレベルで停止する。

3種の文型パターン化では、句レベルの文型パターンを標準とする。これと、具体性が高く高品質の言語変換に適している単語レベルの文型パターンと汎用性が高く被覆率の高い節レベルの文型パターンを組み合わせることによって個性性と汎用性の両立を図る。

2.2 文型パターン化の基準

前節で述べた3レベルの文型パターン化に共通する基準について示す。

(1) 必須要素と任意要素の区分

文型パターン化では、個々のパターンの汎用性を向上させるため必須要素と任意要素を指定する。このうち「必須要素」は日本語文型パターン内にそれが無いと対応する英語文型パターンが決定できない要素である。また、「任意要素」は日本語文型パターン内にその要素がなくても英語文

型は決定可能なもので、パターン定義に使用するか否かにより以下の2つに分類する。

「原文任意要素」：それが削除されても英語生成上も何ら問題のないもの

「パターン任意要素」：それを削除すると、訳語や訳語挿入位置の決定が困難であるなど、その要素自身の翻訳が困難なもの

このうち は、標本文から削除するもので、文型パターン定義には使用しないのに対して、 は、文型パターンに使用される。

なお、「必須要素」と「任意要素」は、いずれも字面でも良いし変数（関数を含む）を含む表現でも良い。

(2) 表現レベルから見た変数化の範囲

日本語表現において変数化する単語、句、節は、原則として客観的な表現の部分とする。テンス、アスペクト、モダリティ、格変化などに関する情報は、「形式指定関数」を使用してパターン化する。

(3) 変数化対象に関する原則

単語、句、節として変数化する対象は、あくまで、線形要素として取り出せる部分であり、変数化した要素の内部が線形構造を持っているかどうかとは無関係である。すなわち、変数化する要素は、原理的にパターンから取り出し、別途翻訳した後、パターンに埋め込めばよい。これより、日本語の文要素を変数化するのは以下の2つの場合となる。

英語側に対応する表現を持つ場合

日英で意味的に対応する部分が同一の品詞（または文法的属性）の場合は、その部分をそのまま該当する変数を使用して変数化する。また、品詞が異なる場合は品詞変換関数を使用し変数化する。

英語側に対応する表現を持たない場合

日本語側、英語側にしか存在しない要素はいずれも原則として字面のままとする。また、あっても良いし、なくても良い要素（「原文任意要素」）はパターンから削除する。

なお 変数化するか、それとも任意化するかの判断は、以下の手順に従う。

- 1) まず、英語に訳されていない日本語表現要素で、それを削除しても英語が変わらないものを探し、「原文任意要素」とする。
- 2) 次に、対訳関係にある表現部分を探す。そのうち、日英の双方からそれを削除しても残りの部分が変化しないものは、「原文任意要素」とし、日英の双方からそれを削除する。残りの部分に影響するものを「変数」とする。

2.3 単語レベル文型パターン化の方法

単語レベルのパターン化では、対訳例文に含まれる「原文任意要素」を取り除いた後、おおよ以下の処理を行う。

(1) 自立語の変数化

表2に示す基準に従って、名詞、動詞、形容詞、形容動詞、副詞を単語変数に書き換える。

(2) 述部語尾表現の関数化

動詞、形容詞、形容動詞に接続する助動詞および助動詞と助詞の連鎖表現（連語）を適切な「関数」を用いて書き換える。なお、この段階では、テンス、アスペクト、モダリティの情報は削除しない。

(3) パターン任意要素の指定

日本語側と英語側の文型パターンを見比べ、「パターン任意要素」の部分に指定する。パターン任意化される要素は、おおむね以下の通りである。

- 1) 助詞結合型名詞句の主名詞を除く部分
- 2) 連体詞、単独形容詞の連体形、単独動詞の連体形

- 3)副詞、単独形容詞の連用形、単独動詞の連用形
4)名詞副詞 5)その他英語に訳出されない要素

表2 自立語の変数化

対象	変数化の内容
名詞	語彙的用法の名詞と複合名詞を名詞変数Nに置き換える。機能語として使用される和語名詞「の、もの、こと、人、とき、場合、原因、理由」などは、変数化しない。
複合名詞	複合名詞を名詞変数Nに置き換える。但し、接頭辞、接尾辞など、それによって英語文型が決定されるような語は変数化しない。
動詞	語彙的用法の動詞の「語幹 + 活用語尾」を動詞変数Vに置き換える。機能語として使用される和語動詞「ある、なる、いる、くる、する」(「漢語動詞 + する」は除く)などは変数化しない。
複合動詞	V+Vの複合動詞は、全体を1変数化せず、「V始める」、「Vでみる」などのように主動詞側を変数化した、アスペクト情報を担う動詞は変数化しない。また、N+Vのような複合動詞は、N、V共に変数化しても良いし、いずれか、両方を字面としても良い。「踏み出す」等は単にVとする。
形容詞 形容動詞	形容詞の変数化は動詞と同様である。また、形容動詞の変数化も形容詞と同じである。「名詞 + 助動詞だ」とはしない。
副詞	名詞の場合と同じである。

(4) 表現要素のグループ化

作成した文型パターンの中で、助詞、助詞相当語、または、副詞などの字面の表現部分が意味的に同じであり、それを除く他のパターン要素がすべて一致するものを対象に日本語文型パターンを縮退させる。

(5) 各種加工

前節までで得られた日英文型パターンに対して、「ゼロ代名詞の補完」、「冠詞の削除」などを行う。

主語の補完

対訳例文において英語パターンで主語となる名詞が日本語側に存在しない場合、日本語側パターンのゼロ代名詞の部分にパターン任意要素の形式で主語を補完する。また、英語側のパターンは該当する箇所に対応する変数を挿入する。

冠詞 (a, the) の削除

英語文型パターンに対して、文の意味上、定まった冠詞を要求する文型では、変数Nの前の冠詞を字面のまま残し、それ以外の場合は、冠詞はすべて削除する。なお、削除された場合の冠詞の決定は、英文生成系に任せる。

表3 句レベルの汎化の方法

種類	説明
現在形 変換	時制に関する表現を現在形に書き換える。但し、それに伴って英語側の文型構造が非線形的に変化するものは対象外とする。
丁寧表現 の標準化	英語文型パターンに影響のない日本語文型パターンの丁寧表現をフラットな表現に変更する。
変数の 適用範囲 の拡大	名詞変数、動詞変数のうち句に拡張しても良いものを選び名詞句変数、動詞句変数に書き換える。なお、連体修飾部を持つ名詞は、全体を名詞句変数に置き換える。また、動詞句変数への書き換えでは、アスペクトとモダリティに関する表現は、削除しない。
機能語 の拡充	格助詞、格助詞相当語の部分と述部の語尾表現で置き換え可能な単語もしくは連語を追加する。
品詞変換 関数の 適用	日本語側の文型パターン要素が、英語文型パターンで文法的カテゴリの異なる要素に対応するものを対象に、変数関数を使用したパターン化を行う。対象となる表現部分は、日本語語側の名詞、動詞がそれぞれ英語側で名詞、動詞になるものである。

2.4 句レベル文型パターン化の方法

すでに述べたように、句レベルの文型パターン化では、

標準的なパターンの作成を目指す。そのため、単語変数から句変数への適用範囲の拡大を図ると共に、英語文型を変化させない範囲での文型の平板化をはかる。なお、アスペクト、モダリティに関する情報は、英語文型を決定する性質が強いので削除しない。汎化の主な内容を表3に示す。

2.5 節レベルの文型パターン化の方法

最も汎用的で、複文、重文の骨格が浮き彫りになるような文型パターンを作成する。汎化は以下の条件を満たす節を対象とする。

必須要素又はパターン任意要素として主語を持ち、英語側で対応する節変数が使用可能なこと

日本語側の節の意味が英語側の節の意味に過不足なく対応すること

また、節変数化される対象は命題レベルの単文(核文とも言う)表現部分とし、アスペクト、モダリティの情報は文型パターンから削除しない。

3. 文型パターン記述言語の設計

第2章で述べた文型パターンを記述するための言語(以下では、「文型パターン記述言語」または、単に「記述言語」とも言う)の基本仕様について以下の条件で検討する。

(1) 3レベルの文型パターンが記述できること

(2) 日英で線形部分の対応関係が明確であること

(3) 読みやすいこと

日本語文型パターンと英語文型パターンは「字面」、「記号」、「変数」、「関数」の4種類で表記される。このうち、「字面」は、日本語文型パターンでは、仮名、漢字、数字など(EUC code)であり、英語文型パターンでは、アルファベット、数字など(ASCII code)である。以下、本節では、「記号」、「変数」、「関数」の仕様の概要を示す。

3.1 文型記述に使用される記号

文型パターン記述言語で使用される記号は、表4に示す4種類である。このうち、「離散記号」と「連鎖記号」は、入力原文と文型パターンを照合する際、入力文中に当該要素が連鎖して出現しなければならないか否かを表すものである。また、「要素選択記号」は、日本語文型パターン内の当該位置に使用可能なパターン要素が複数存在することを表し、「任意要素記号」は、省略可能要素であることを表す。

日本語文型パターンに「任意要素記号」が使用されている場合、対応する英語文型パターンでは、この記号が使用される場合と使用されない場合がある。但し、使用されるのは、「#n[]」の記法のみで、複数要素からの選択はできない。この場合の日英文型パターンでの意味的な対応関係は以下の通りと解釈される。

日本語文型パターンで使用された任意要素記号が英語文型パターンで使用されないときは、日本語側の当該表現は英語の訳出する必要がない

日本語文型パターンで使用された任意要素記号が英語文型パターンで使用されているときは、日本語入力文に該当する要素がある場合にのみ、その英語訳を必要とし、日本語入力文に該当する要素がない場合は、その英訳も不要である。

3.2 文型記述に使用される変数

変数は、文型パターンの中で置き換え可能な部分的表現を表す記号である。表5に示すように、表される部分的表現の文法的属性に対応して、単語、句、節を表す変数に大別される。いずれもアルファベット大文字と数字で表現される。数字は日本語文型パターン内での変数の出現順を意味

表4 パターン記述に使用される記号（日本語側のみで使用）

分類	記法	引数	機能説明	備考
離散記号	/	なし	原文任意要素がゼロ個以上存在可能な文節境界を意味する	
連鎖記号	+（省略可）	なし	原文任意要素が存在してはいけない文節境界を意味する	
要素選択記号	(arg1 arg2 ...)	字面、変数、関数を含む任意の文字列	括弧で囲まれた要素は代替可能な要素で、いずれか一つが使用されることを意味する	
任意要素記号	#n[arg1 arg2 ...]	字面、変数、関数を含む任意の文字列	指定した文字列は入力日本語文中にあってもなくても良いことを意味する。#nは、パターン中での該要素の出現番号。また、引数は、いずれも意味的に代替可能な表現。	注1 注2
<注1>主語や目的語の補完の機能もこの仕組みを使用して記述する。 <注2>日本語文型パターンで指定された要素が入力文にあり、且つ、英語文型パターン内にもある場合は番号で対応関係を取る。また、日本語文型パターンで指定された要素が入力文にあるが、対応する要素が英語文型パターンに存在しないときは、英語表現では無視される。				

表5 文型パターン記述に使用される変数

分類	変数名	機能説明
単語	Nn	名詞又は名詞複合語を表す
	Vn	動詞の「語幹+活用語尾」を表す。
	AJn	形容詞の「語幹+活用語尾」を表す。
	AJVn	形容動詞の「語幹+活用語尾」を表す。
	ADVn	副詞を表す
	TIME _n	時間を表す
句	VP _n	動詞句の原型をあらわす
	NP _n	名詞句の原型を表す
節	CL _n	節を表す。但し、時制、相、様相なしの現在形で核文を表す。
<備考1>すべての変数には、パターン内の通し番号nが付与される。 <備考2>用言の変数の場合は原型がバインドされる。		

する。

日本語入力文と日本語文型パターンの照合に成功したときは、変数には該当する日本語表現の部分がバインドされる。すなわち、変数にその値として日本語表現が代入される。また、その変数が英語文型パターンで使用されたときは、その変数はバインドされている日本語表現の英語訳を意味する。

なお、動詞などの活用語では、日英とも変数にはその原形がバインドされる。

例) 日本語側：N1= 私達、V2=食べる、等
 英語側： N1= we、 V2=eat、 等

また、英語文型パターンでは、名詞変数は原則として変数は冠詞を含む名詞を意味するものとする。

3.3 文型記述に使用される関数

文型パターンの記述では、「変数関数」、「字面関数」、「要素抽出関数」、「形式指定関数」、「構文合成関数」、「マクロ関数」の6種類の関数を使用する。

これらが日本語側で使用された場合は、該当する部分的表現が持つべき形式を表し、入力日本語との照合に成功すれば、引数で使用された変数に該当する表現がバインドされる。また、英語側で使用されると該当する部分で関数で定義された英語表現が生成される。

関数は、マクロ関数（後述）を除き、いずれも多重使用（関数引数として関数を使用すること）を可能とする。また、関数の引数には、字面、変数、関数を含む表現が使用されるが、表現の属性は関数毎に定められた属性（「単語」、「句」、「節」のいずれか）を持たなければならない。以下、各関数の概要を示す。

3.3.1 変数関数

変数名が関数名として使用される関数で、表6で示すように、動詞、形容詞、形容動詞を名詞に変換するもの、

表9 形式指定関数の種類

名詞を動詞に変換するもの、動詞句、節を名詞句に変換するもの、名詞句を動詞句に変換するもの、等がある。

表6 変数関数の種類（英語文型パターンでのみ使用）

分類	関数形	引数	機能説明
文法属性変換	N(arg)	V, AJ, AJV	動詞、形容詞、形容動詞を名詞に変換する
	V(arg)	N, AJV	名詞、形容詞を動詞に変換する
	NP(arg)	VP, CL	動詞句、節を名詞句に変換する
	VP(arg)	NP	名詞句を動詞句に変換する
	NP _{pred} (arg)	CL, VP	argの述部に焦点を当てた名詞句に変形する。
	NP _{subj} (arg)	CL, VP	argの主語に焦点を当てた名詞句に変形する。
	NP _{obj} (arg)	CL, VP	argの目的語に焦点を当てた名詞句に変形する。

いずれも英語文型パターンでのみ使用される関数で、引数で指定された表現を関数名で指定された文法的属性の表現に変換する。

例えば、N(Vn)では、日本語側の変数Vnにバインドされた表現は、「日本語動詞 英語動詞 英語名詞」の順序での変換が試みられるが、これができないときは、「日本語動詞 日本語名詞 英語名詞」が試行される。いずれにも失敗したときは当該パターンは使用されない。

3.3.2 字面関数

関数名として字面を使用した関数で、日本語文型パターンでのみ使用される。表7に示すように、引数で指定された表現が関数名で示される字面を持つことを意味する。但し、原文との照合の結果、引数とする変数には、その字面を除いた表現がバインドされる。

例えば、「外は大変暑かったので」は、「past(大変(CL1))」なのでとの照合に成功し、CL1=「外は暑い」がバインドされる。

表7 字面関数の仕様（日本語側でのみ使用）

分類	関数形	引数	機能説明
字面関数	字面(arg)	VP, CL	関数名に字面を使用する関数で、引数表現が関数名で指定された字面を含むことを表す。引数には関数名部分を除く表現がバインドされる。

表8 要素抽出関数（英語側でのみ使用）

分類	関数形	引数	機能説明
文要素の抽出	subj(arg)	CL	argの主語を取り出す。
	obj(arg)	CL	argの目的語を取り出す。

関数種別	日本語文型パターンのみで使用	日英両文型パターンで使用	英語文型パターンでのみ使用
語形指定関数	断定： <i>da</i> (arg) / <i>danntei</i> (arg)、丁寧： <i>polite</i> (arg)、謙譲： <i>kenjo</i> (arg) / 不興事態： <i>unfav</i> (arg) / 過剰： <i>sugiru</i> (arg) / ている： <i>teiru</i> (arg) / てくる： <i>tekuru</i> (arg) / てある： <i>tearu</i> (arg) / ていく： <i>teiku</i> (arg) / 状態変化： <i>naru</i> (arg) / 伝聞、他： <i>souda</i> (arg)	過去： <i>past</i> (arg) / 可能： <i>can</i> (arg) / 否定： <i>not</i> (arg) / 不可能： <i>cannot</i> (arg) / 許可： <i>may</i> (arg) / 義務： <i>should</i> (arg) <i>must</i> (arg) / 推量： <i>seem</i> (arg) <i>darou</i> (arg) / 重畳語： <i>dual</i> (arg) / 良好： <i>better</i> (arg) / 不可避： <i>fukahi</i> (arg) / 開始： <i>begin</i> (arg) / 予定： <i>yotei</i> (arg)	目的格変形： <i>obj</i> (arg) / 所有格変形： <i>poss</i> (arg) / 独立所有格変形： <i>abposs</i> (arg) / 再帰代名詞変形： <i>reflex</i> (arg) / 複数形変形： <i>pl</i> (arg) / 名詞、名詞句を代名詞に変換： <i>pron</i> (arg) / 完了形変形： <i>pti</i> (arg) / 現在分詞変形： <i>ing</i> (arg) / 過去分詞変形： <i>ed</i> (arg) / 動名詞変形： <i>grn</i> (arg) / 使役形を能動態に変換： <i>uncause</i> (arg) / 自動詞を他動詞に変換： <i>intrans</i> (arg) / 他動詞を自動詞に変換： <i>trans</i> (arg) / 最上級変形： <i>st</i> (arg) / 比較級変形： <i>er</i> (arg) /
構文指定関数	態の変形： <i>passive</i> (arg)	意志： <i>will</i> (arg) / 受け身： <i>passive</i> (arg) / 受益： <i>kureru</i> (arg) / 被害受身： <i>suffer</i> (arg)	受動態変形： <i>passive</i> (arg) / 「強調語 + V+S」の文型に変形： <i>inverse</i> (arg) / 「強調語 + as+V+S」の文型に変形： <i>inverse_as</i> (arg)
特殊			驚き表現「To Sの所有格 + Vの名詞形」の形式に変換： <i>to_emotion</i> (arg)

表10 構文合成関数（英語文型パターンでのみ使用）

分類	関数形とその機能
分詞構文	<i>ptcle</i> (CL/VP)：引数を分詞構文に変形する /
使役変形	<i>cause</i> (N, V)：使役形に変換 / <i>uncause</i> (N, V)：使役形を通常形に変換 /
因果関係構文	<i>so_that</i> (CL, CL)：so that構文を合成する。 / <i>so+that</i> (CL, CL)：同前。但し引数1の末尾にカンマを入れる。 / <i>such_that</i> (CL, CL)：such that 構文を合成する / <i>such+that</i> (CL, CL)：引数1の補語をsuchにし、引数1と引数2をthatで結ぶ
因果の程度表現	<i>too_for_to</i> (CL, NP, VP)：引数1にtooを、引数3の前にforをつけ引数2を不定詞変形 / <i>too_for_even+to</i> (CL, NP, VP)：引数2の不定詞形にevenをつける他は上と同じ / <i>much+too_for_to</i> (CL, NP, VP)：too_for_toの構文で、tooをmuch too変えるもの / <i>so_as+tor</i> (CL, NP, VP)：引数1、引数3の前に各so、forをつけ引数2を不定詞変形 /
驚き表現	<i>to_emotion</i> (CL)：引数を「To Sの所有格 + Vの名詞形」の形式に変換

表11 マクロ関数

分類	記法	引数	機能説明	
			日本語文型パターン	英語文型パターン
マクロ関数	CL#n(arg) NP#n(arg) VP#n(arg)	それぞれ、節構造、名詞句構造、動詞句構造を持つ表現パターン	引数で示された表現をそれぞれ変数CL#n、NP#n、VP#nにバインドする。引数の中で使用された変数には、該当する表現がバインドされる。	CL#n、NP#n、VP#nを変数名として使用する

3.3.3 要素抽出関数

英語文型パターンでのみ使用される関数で、表8に示すように、引数にバインドされた表現から特定の文要素を取り出す。

例) CL1=We eatanapple のとき、*subj*(CL1)=we

3.3.4 形式指定関数

形式指定関数は、動詞や動詞句、節の過去形、可能形、否定形、受身形などを表す関数で、日本語文型パターンでは、入力文表現に対する適合条件として働き、英語文型パターンでは表現生成の生成条件として働く。

この関数は、表9に示すように、日本語文型パターンの記述のみに使用されるもの、日英両文型パターンの双方で使用されるもの、英語文型パターンでのみ使用されるものの3種類に分けられる。

このうち、この関数では、日本語入力文での照合に成功すると、引数で示された変数には、関数名で示される情報を取り除いた原文表現がバインドされる。また、この場合は、英語文型パターン側に引数に相当する表現の訳は陽には現れないのに対して、この場合は、引数は日英両文型パターン間で対応が取られる。なお、この場合は、日本語文型パターンには同一の関数は使われていないが、引数は日本語文型パターンで使用されていることが必要である。

3.3.5 構文合成関数

構文合成関数は、引数で指定された複数の表現から関数名で指定された構造の表現を合成する関数で、*so_that*構文、*too_for_to*構文、*so_as+to*構文など関数名で指定された表現を合成する。英語文型パターンでのみ使用される。この関数で使用される引数には、それぞれの関数で定められた要件を満足する表現がバインドされていることが必要である。その種類と機能を表10に示す。

3.3.6 マクロ関数

変数名を関数名とする関数で、変数にバインドされる表

現の構造を定義する関数である。関数名としては、句変数と節変数が使用され、日本語文型パターンでのみ使用される。形式と機能を表11に示す。

対象とする表現が引数で指定された構造を持つとき、その表現が関数名として使用されている変数の値となる。対象とする表現が引数で指定された構造を持たないときは、照合に失敗する。使用例を下記に示す。

和文P：*past*(CL#1(N1はADV1AJV1))の*past*(*cannot*(CL2))
= N1はADV1 *past*(AJV1) ので*past*(*cannot*(CL2))
適用例：「その星は大変暗かったので、我々は見る事ができなかった。」
英文P：*so_that*(CL#1, CL2)
= N1was so AJV1 that N2VP1.
適用例：The star was so dark that we could not see it.

3.4 文型パターン記述例

前節までで定義された方法を使用して、「単語レベル」、「句レベル」、「節レベル」で汎化されたパターンの記述する例を示す。

(1) 単語レベルの汎化例

和文P：#1[N1は]V2て/N3を/N4に/*tekita*(V5)。
英文P：It was so AJ(V2) as to V5#1[*poss*(N1)] N3atN4。
和文) うっかりして定期券を家に忘れてきた。
英文) It was so care less as to leave my season ticket at home.

英語側のAJ(V2)は、日本語V2の形容詞形を使用することを意味する。また、入力日本語文には、英語で主格に相当する「は格」(名詞N1)はなくても良いが、それがある場合には、英語文型では、N3の冠詞の代わりにN1の所有格を使用することを示す。

(2) 句レベルの汎化例

和文P：NP1は/*past*(V2)/N3に/*teiru*(V4)のだから/*da*(AJ5)。
英文P：NP1 is AJ5 in that N1 V4 on AJ(V2) N3。
和文) その結論は誤った前提に基づいているのだから誤りである。
英文) The conclusion is wrong in that it is based on false premises.

名詞句変数NPが使用されている。動詞V2、動詞V4はそれぞれ「過去形」、「ている形」が指定されている。また、英

語側では、動詞V2の形容詞形が使用されている。すなわち、動詞V2の形容詞形が存在しないときは、このパターンは使用されない。

(3) 節レベルの汎化例

和文P: CL1ので、N1にあたっては/must(VP2)

英文P: so+that(CL1, passive(must(VP2)))with

poss(subj(CL1) N1)

和文)これは極めて有毒であるので、使用に当たっては十二分に注意しなくてはならない。

英文)It is significantly toxic so that great caution must be taken with its use.

英語文型パターンでは、節CL1と動詞句VP2から、so that構文を生成することを示している。

4. 入力日本文との照合基準

文型パターン作成規則と入力文に対する文型パターンの適用方法は相互関連が大きい。本章では、パターン適用法の基本的な考え方を示す。

4.1 適合パターンの基本的要件

日英文型パターンは共に「必須要素」と「任意要素」から構成される。日本語入力文には、以下の条件を満たす日本語文型パターンが適用される。

入力文はパターンで定義されたすべての「必須要素」を持ち、その制約条件を満たさねばならない。

「パターン任意要素」は持たなくても良いが、それを持つ場合は、その制約条件を満たさねばならない。

入力文は文型パターンで定義された文節要素間に原文任意要素として任意の文節要素を持つことができる。

入力文における文型パターン構成要素(「パターン任意要素」を含む)の出現順序は、パターン内での出現順序と一致しなければならない。

4.2 パターンの適用順序と入力文の解析

(1) パターンの適用順序

入力文に対する文型パターンの適用順序は以下の通りである。

パターンは、汎化の少ないレベルから順に適用する。適用可能なパターンが発見されれば、それ以上のレベルでのパターンは検索しない。

同一レベルで適用可能なパターンが複数存在する場合は、パターン内に定義された要素数が大きいものを優先的に使用する。

なお、入力文に対して同時に複数のパターンを適用する場合も考えられるが、それについては規定しない。

(2) パターン照合のための入力文の解析

入力文と文型パターンとの照合では、品詞情報、文節境界情報を得るため、入力文の形態素解析結果を用いる。

4.3 パターン照合の方法

(1) 標準時制変形によるパターン照合

入力文と単語レベルの文型パターンとの照合では、入力文の時制の変更は不要で、そのままの時制で照合して良い。

(2) 文型変形によるパターン照合

入力文が受動態、疑問文、使役文などのような場合は、一度、そのままの形式でパターン辞書と照合する。その結果、適切なパターンが得られないときは、入力文を平叙文形式に変換した後、改めて、パターン辞書と照合するものとする。また、平叙文形式の入力文でも、格助詞相当語を格助詞に置き換えたあとパターン照合することもかまわない。

(3) パターンの拡大適用の可否

パターンの変数部分について拡大解釈による適用を認めない。すなわち、名詞変数Nに名詞句を適用したり、副詞ADVに副詞句を適用したりしてはならない。

(4) 語順変更によるパターン適用

日本語文型パターンでは、文型要素の順序変更の可否は定義しない。従って、各要素間の語順変更や述部と各要素との語順変更の可否については、パターン適用側の判断にゆだねる。

(5) 読点の照合

日本語文型パターンでは、必要な読点は必ず記載されるから、パターン照合では、まず、読点を含め要素列の一致する文型パターンを検索する。それが得られないときは、文節境界に存在する入力文の読点、及び、文型パターンの読点は、それぞれ、「原文任意要素」、「パターン任意要素」として扱う。

5. むすび

等価的類推思考の原理による機械翻訳方式の実現に必要な「意味類型知識ベース」を構築するため、日英両言語の対訳例文から文型パターンを収集する方法について論じ、両文型パターンを記述するための文型パターン記述言語を提案した。

文型のパターン化では、パターン間の相互独立性を保ちつつ、パターン全体で高い被覆率を実現することが重要である。そこで、本稿では、「単語レベル」、「句レベル」、「節レベル」の3つのレベルを定め、大量の対訳コーパスから順に汎化を行うことによってこれらのパターンを作成する方法を考えた。

今後、対訳例文からの文型パターン化に適用すると共に、「日本文との照合アルゴリズムとその実現性」、「英文生成アルゴリズムの実現性」について検討していく予定である。また、本稿では触れなかったが、パターン間で干渉が生じる場合、変数に対して意味的な制約条件が指定できるようパターン記述言語の拡張を行う予定であり、ここで提案した変数、関数は、これらの過程で修正補充される可能性がある。

< 謝 辞 >

この研究は、科学技術振興事業団(JST)の戦略的基礎研究事業(CREST)で行っているものである。

< 参 考 文 献 >

- [有田87] 有田潤:「ドイツ語講座」南江堂(1987), pp.48-56
- [池原87] 池原悟, 宮崎正弘, 白井論, 林良彦: 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, Vol.28, No.12, pp.1269-1279 (1987)
- [池原97] 池原悟, 宮崎正弘, 白井論, 横尾昭男, 中岩浩巳, 小倉健太郎, 大山芳史, 林良彦:「日本語語彙大系」岩波書店(1997)
- [池原01] 池原悟: 自然言語処理の基本問題への挑戦, 人工知能学会誌, Vol.16, No.3, pp.522-430 (2001)
- [Ikehara 01] Satoru Ikehara: Meaning Comprehension Using Semantic Patterns in a Large Scale Knowledge-Base, Proceedings of the PACLING'01, pp.26-35(2001)
- [池原02] 池原悟: 究極の翻訳方式の実現に向けて = 類推思考の原理に基づく翻訳方式 =, AAMT Journal, アジア太平洋機械翻訳協会, No.32, pp.1-7 (2002)
- [市川63] 市川亀久彌:「創造的研究の方法論」(増補版), 三和書房, 1963
- [金出地01] 金出地真人, 池原悟, 村上仁一: 結合価文法による動詞の訳語選択能力の評価, 情報処理学会第63回全国大会, 6Y-04, 2-267-268(2001)
- [三浦67] 三浦つとむ:「言語と認識の理論」第1~3巻, 勁草書房(1967)