オブジェクト指向言語のパラダイムを利用した機械翻訳エンジン jaw

今井啓允、池田尚志
岐阜大学工学部

1　はじめに

我々は、パターン変換型機械翻訳エンジン jaw(from Japanese to Asian and world languages) の開発を行っている。jaw は日本語から他の言語への翻訳を行うもので、現在アジアの諸言語 (中国語、シンハラ語、ベトナム語、ミャンマー語) および手話 (日本語テキストからの手話単語列) への機械翻訳を試みている。jaw は日本語文の解析、表現パターン辞書との照合による相手言語の表現構造への変換、表現構造からの相手言語の生成といういわゆるトランスファーの方式である。翻訳エンジンは同一で、変換規則、生成関数が相手言語ごとに異なっている。

システムの特徴は命題的内容の翻訳、用言後接機能語部分の翻訳、体言後接機能語の翻訳と翻訳を 3 段階に分けて翻訳を行うこと、オブジェクト指向言語のパラダイムを利用していることなどである。未だ初期的なパイロットモデルの段階であるが、本稿では命題的内容の翻訳、用言後接機能語の翻訳処理方式、最適解の選択等について述べる。

2　機械翻訳エンジン jaw の概要

jaw は、日本語と目的言語の対応規則を日本語文の係り受け木構造のパターンとそれに対応する目的言語の表現構造の対（表現パターン辞書）という形で表現している。まず日本語入力文を文節構造・構文解析して日本語の係り受け構造 IT(InputTree) を作る。次に IT と表現パターン辞書とを照合し、IT を日本語パターンと対応する翻訳規則の木構造に変換する (TT:TransferTree)。次に TT の中の翻訳規則に対応づけられた翻訳プログラム（dll として実現している）を実行することで目的言語の表現構造ネットワーク (ET:ExpressionTree) を生成する。ET は、VC++のオブジェクトとして実現している。最後に、各 ET に対応した生成関数を実行することで目的言語の文を生成する。生成関数は VC++の各クラス（動詞クラス、名詞クラスなど）に定義されているクラスメソッドである。jaw では、目的言語ごとに、その表現構造のためのクラスを設計し、クラスメソッドである生成関数（線状化関数）を定めれば目的言語に対する翻訳システムが構築できる。個々の翻訳知識は表現パターン辞書および機能語翻訳規則辞書中に記述されることになる。(図 1)

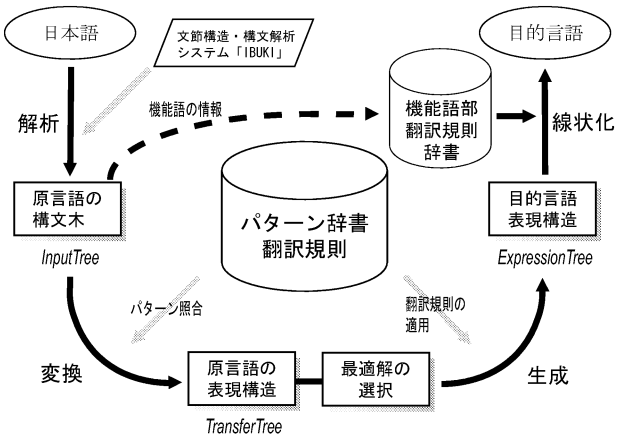

図 1: 機械翻訳システム jaw

表現パターン変換辞書との照合による TT の作成、TT に対応づけられた翻訳規則の実行による ET の生成、機能語部翻訳規則データの ET への付加、ET に対応づけられた生成関数の実行による目的言語の生成という方式は、翻訳知識データとその処理を完全に分離しており、任意の言語を目的言語として対象とすることができる。

3　表現パターンの照合

jaw のパターン辞書翻訳規則 DB 中の表現パターンは、記述の中心となる語（キーワード）を持つ。キーワードの種類やキーワードを持つ文節に係る文節などの条件から、表現パターン辞書は Base Type、Addition Type の 2 種類に分けられる。表 1 にこれらの例を示す。

Base Type はキーワード文節にどのような文節が係るかを記述したパターンであり、Addition Type はキーワード文節がどのような文節に係るかを記述したパターンである。表 1 で AddCW は Addition Type

でキーワードが自立語の場合、AddFW はキーワードが機能語の場合である。

表 1：表現パターンの例

Type	文節番号	係り先番号	KW	自立語条件	機能語条件
Base	1	3		人 (彼)	「が」
	2	3		具体 (男)	Null
	3	0	だ	−	−
Base	1	3		人 (私)	「が」
	2	3		人 (彼)	「と」
	3	0	付き合う	−	−
Base	1	0	彼	−	−
Base	1	0	男	−	−
AddCW	1	2	面白い	−	Null
	2	0		具体 (男)	−
AddFW	1	2	てみると	動作	−
	2	0		状態	−

表現パターンには、対応する ET（VC++のオブジェクト）を生成するプログラム（DLL）が対応付けられており、IT と表現パターンとの照合の結果として、対応する翻訳規則の木 (TT) が作られる。

「彼と付き合ってみると面白い男だった。」を照合したときの TT を図 2 に示す。

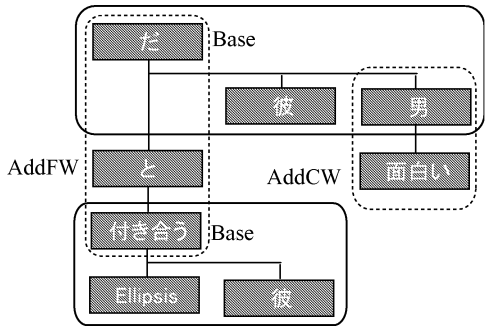

図 2: 翻訳規則の木

これらの詳細については、[2] で述べた。以下本節では、連体埋め込み表現、慣用表現、字づら照合などのいくつかの特殊な照合処理について述べる。

・受身、使役文の照合

「食べさせる」「騙される」といった受身・使役の表現パターンは、パターン変換辞書には登録せず、通常の文から受身使役文に変わったときに起こる格の交代に着目し、機能語条件の読み替えを規則化することで、従来の表現パターンから受身・使役のパターンを作成する。

・連体埋め込み表現の照合

「熊が歩く道」「道を歩く熊」など修飾文節が外に出ることで連体修飾構造になる表現は多くの用言によく起こり全てを表現パターンとして登録はできない。

そこで、連体埋め込み表現の照合は通常の Base Type の用言パターンが変化したものとして、連体修飾構造の表現パターンを作成することで照合を可能にしている。

・慣用表現、字づらの照合

「道草を食う」などの慣用表現は「＜食料＞を 食う」というパターンとは、意味合いが違う。そこで、「[道草] を 食う」パターンを作成し、意味属性ではなく [道草] という字づらで照合を行うことで、慣用表現も正しく照合ができる。

また、慣用表現には「＜人＞の [口] に＜食料＞が合う」といった深さが 2 以上となるパターンも多く存在する。深さ 2 以上のパターンによって TT が生成された場合は、深さ 2 以上の位置にあるノードはリンクを変更することで、翻訳規則によるオブジェクトの操作を簡単にしている。(図 3)

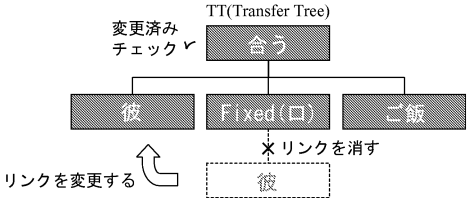

図 3: リンクの変更

・複合語の照合

複合語は複合語解析によって、単語ごとに区切られる。

弁論大会 → 弁論 /大会
田中君 → 田中 /君

複合語である場合は日本語の構文木である IT を単語ごとにノードに分割し、さらに「_複合語」という仮のノードを親ノードとすることで複合語で表現した。(図 4)

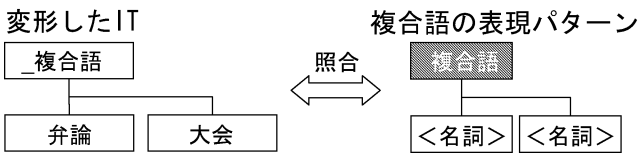

図 4: 複合語の照合

生成された複合語の TT をどのように翻訳するかは、目標言語ごとの今後の課題である。

4　最適解の選択

IT から照合によって生成された TT については、多く場合に複数の解が存在する。よって、その中から最適であると思われるものを選択する必要がある。最適解を選択するために以下の 3 つの条件を考えた。

- TT を構成するパターンの数や種類による条件 (1)
- 適用されたパターンの厳しさによる条件 (2)
- 適用された意味属性同士の距離による条件 (3)

(1) の条件は、TT を構成するパターンの数が少なければ少ないほど最適であるという条件である。また、補助的な Addition Type よりも TT の骨格となる Base Type のほうがより最適である。

(2) の条件は、TT に適用された表現パターンの条件が厳しいものほど最適パターンと考えるものである。表現パターンが持つ自立語や機能語の条件から意味属性や機能条件の出現頻度などから各パターンの厳しさを判定する。

(3) の条件は、自立語条件に適用されたパターンが複数ある場合どのパターンの意味属性が自立語条件が持つ意味属性に近いかを判別することで最適なパターンを選択する。以上 3 つの条件を総合してコストを計算し最もコストの低いものが最適な解となる。

5　C++オブジェクトを用いた目的言語の表現構造

表現構造 (ET) は目的言語の文の一種の意味表現である。我々はそのために VC++のクラスオブジェクトを利用した。名詞、動詞などのクラスを定義し、中心語とそれを修飾する表現など、表現のための部品をクラスメンバーとして定義しておく。また、そのインスタンスが与えられたときに、これらの部品を線形につなぎ合わせて目的言語の 1 次元の言語表現、つまり翻訳文を作り上げるための関数（生成関数あるいは線状化関数と呼んでいる）と、クラスのメンバー関数（メソッド）として定義しておく。これらのクラスの設計（どんなクラスを設けるか、どんなメンバー変数を設けるか、どんなメソッドを与えるか）は、目的言語毎にその文法構造に応じて設計することになる。クラスとそのメンバの例を表 2 に示す。

表 2 : クラスとメンバの例

クラス	メンバ	説明
CProposition 動詞・形容詞	m_centerW	クラスの中心語 (訳語)
	m_subject	主格 (CNoun)
	m_object	目的格 (CNoun)
	m_nounModifier	その他の格 (CNoun)
	m_pConnection	動詞との接続
CNoun 名詞	m_centerW	クラスの中心語 (訳語)
	m_casemarker	前置詞
	m_role	格の役割
CpConnection 接続詞等	m_connection	接続詞等の訳語
	m_pSubbordinate	接続先の動詞

日本語パターンに対応する目的言語の表現として記述された翻訳規則はデータベース上のテーブルに表現されており (表 3)、TT から ET を作るプログラム (翻訳規則関数) に変換される。これはコンパイルされて翻訳規則のライブラリ (dll) となる。

表 3 :「だ」「彼」「男」の翻訳規則

Type	Class	mClass	Member	Value
Base	CProposition	CNoun	m_subject	#1
−	CProposition	CNoun	m_object	#2
−	CProposition	CString	m_centerW	是
Base	CNoun	CString	m_centerW	他
Base	CNoun	CString	m_centerW	男人

この翻訳規則を記述・編集するための編集システムを作成した (図 5)。表現パターンの新規登録・更新、翻訳規則の記述・編集やパターンの検索などができる。さらに、作成した翻訳規則から、ET を作り出すための VC++のソースプログラムを出力する機能を備えている。

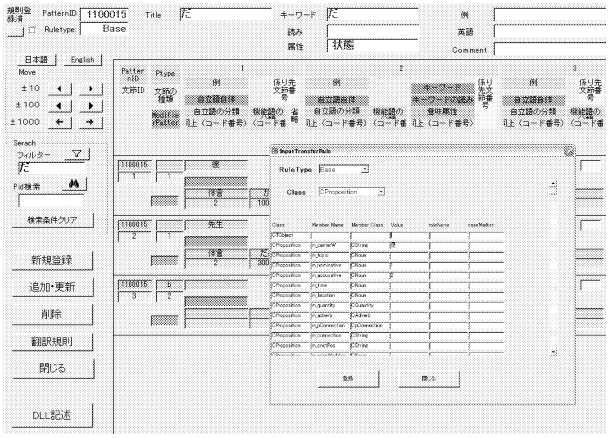

図 5: 翻訳規則の記述編集フォーム

図 6 に入力文「彼と付き合ってみると面白い男だった。」に対する中国語での ET(VC++のオブジェクトのネットワーク) の例を示す。

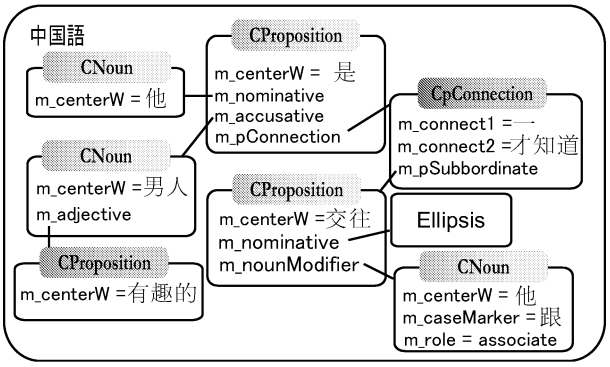

図 6: 中国語の表現構造

6 機能語部分の翻訳

機能語部分は日本語の文法構造の中核をなす重要な部分であり、日本語からの機械翻訳において、それをどう処理するかは重要なポイントとなる。我々は個々の小単位の機能語ごとに翻訳処理を考えるのではなく、文節構造の要素ごとに翻訳処理を対応させる方針をとった。これは、我々が定めた文節構造の各要素の異なり数は、対処可能な有限の範囲に収まるということを仮定したものである。機能語部全体をひとつのまとまりとして翻訳処理対象とすることができれば、さらに対処は容易であろうが、それでは異なり数が膨大になり現実的ではない。しかし、4つの要素に分割して考えれば対処可能であろう、との仮定である。小単位の機能語で考えれば、異なり数はもちろん最小であるが、対処の内容が複雑になり過ぎると考えた。

現在のところ、用言系文節の機能語部分に対応する翻訳規則についての対処を行っており、機能語部要素ごとに機能語の翻訳規則を定め、各要素の翻訳規則を組み合わせることで機能語の翻訳を実現している。表 4, 5, 6 は、中国語の場合の各要素の翻訳規則の簡単化した例である。[4]

表 4 :「使役・受身等」の翻訳規則

機能語	文型	文型前置詞	副詞種類	副詞
られる	受身文	被		
させる	使役文	使		
たい			希望	想
させる/られる			評注性	不得不
がちだ			頻度	常常
させる/やすい	使役文	使	頻度	容易
られる/がちだ	受身文	被	頻度	容易
てもらう	使役文			

表 5 :「時制等」の翻訳規則

機能語	判別処理	副助詞	副詞種類	副詞
た	Ta			
ている	Teisru			
ない	Nai			
てみる			評価	試着
でもある		是	重複	也

表 6 :「判断等」の翻訳規則

機能語	助動詞種類	助動詞	副詞種類	副詞
らしい			評註性	好像
かもしれ/ない	可能性	可能		
しか/ない			評注性	只有

「ない」「ている」などの訳が 1 対 1 に定まらない曖昧性を含む機能語の翻訳はさらに機能語ごとの訳語決定規則を用いることで訳語の曖昧性を解消している。各要素の翻訳規則は目的言語側の機能語情報としてメンバー Mode に格納され、生成関数で適切な処理を受ける。

7 おわりに

オブジェクト指向言語のパラダイムを利用したパターン変換型機械翻訳エンジン jaw の概要について述べた。

今後は複文、重文など大域的なパターンに対応した照合が扱えるよう検討を進めていく予定である。

参考文献

[1] パターン変換型翻訳システム jaw について-、今井、謝、池田、FIT（情報科学技術フォーラム）2002,E-44

[2] 日本語からアジア諸言語への機械翻訳の試み、今井、謝、T.Samantha、酒井、高木、E.Nayan、M.Chau、卜、池田、情報処理学会、第６５回全国大会 (2003) p5-363 - 5-366

[3] 日中機械翻訳における中国語語順の決定法について、謝、今井、卜、池田、FIT（情報科学技術フォーラム）2002

[4] 軍、今井啓允、池田尚志：日中機械翻訳システム jaw/Chinese における変換生成の方式、言語処理学会：自然言語処理　第 11 巻　第 1 号 p43-80、(2004.1)

[5] An Approach to Translate Japanese Modality into Chinese Exressions Xie、Imai、Bu、Ikeda、ICCPOL(2003)