

機械翻訳システム Jaw における機能語独立文節に関する処理

—Jaw/Chinese における事例—

薛明恵, 池田尚志

岐阜大学工学部

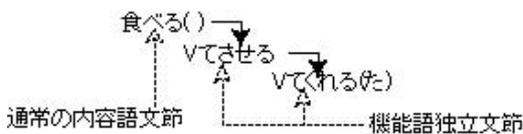
1 はじめに

我々は機械翻訳において、用言後接機能語の処理が重要であると認識している。我々の研究室で開発している日本語から各国語への翻訳システム Jaw において“機能語独立文節”という概念を導入し、それに基づいて用言後接機能語処理を更に展開した。

2 機能語独立文節

機能語独立文節[1]とは、用言後接機能語要素を用言文節から独立させ、用言文節と同じレベルで扱うものである。

例文：私たちにいい料理を食べさせてくれた。



各文節の構成：内容語（機能語要素）

図 1 - 「食べさせてくれた」の解析結果

上図の示すように、機能語「させる」は機能語独立文節「Vてさせる」として扱い、機能語「くれる」は機能語独立文節「Vてくれる」として扱うこととした。

このような機能語独立文節との組み合わせによって、機能的な表現を内容語と同様の扱いで処理することにした。

3 機能語独立文節のパターン

Jaw では機能語独立文節の内容語をキーワードとするパターンに二つの種類がある。

**Fn 型** : 用言に意味を追加する。

**Fw 型** : 文節と文節を接続する。

3.1 Fn 型パターン

Fn 型パターンは機能語独立文節の内容語をキーワードとして、それが用言文節を受けるという形のパターンである。

V(fn\_0) F(fn\_1)

注：V は用言；F は機能語独立文節の内容語；fn\_0,fn\_1 は機能語要素で、V or F に後接する。「た、ない」など。

- ① 必要な V の意味属性は翻訳規則に記述する。
- ② fn\_0,fn\_1 などの機能語要素は必要に応じて、この Fn 型パターンの一部に含めその翻訳規則で処理することも出来るし、従来どおりにパターンと分離して機能語翻訳規則で処理する[2]ことも出来る。

```

{
  ... ..
  bool bSetValue = false;
  TCHAR atcValue22_1_1[] = {49, 50, 55, 48, 48, 0}; // 12700
  if( bSetValue == false
      && OpIsNotNull(var->getNModi(_T("time")))
      && OpEqual((LPCTSTR)(b_str_t)(var->getNModi(_T("time")))-
                >m_model.m_semanticAttri,atcValue22_1_1))
      { bSetValue = true;
  }
  ... ..
  if( bSetValue == false && OpDefault())
  { TCHAR atcTrWord100_1_1[] = {0};
    var->m_modeC->aspect_teiru = atcTrWord100_1_1;
    TCHAR atcTrWord100_1_2[] = {115, 101, 108, 102, 0}; // self
    var->m_model.m_UsedFunctionWord_ReUse = atcTrWord100_1_2;
    bSetValue = true;
  }
  return var;
}
    
```

図 2 - 翻訳規則関数

例として「Vている(た)」に対する翻訳処理を示す。

パターン： V() Vている(た)

{V() Vている(た)}に対する翻訳規則：

- ① 時点を表す時間副詞がVに掛かれば、「Vている(た)」に関する訳語を無しにする。
- ② そうでなければ、「Vている」に関する訳語を無しにし、機能語要素「た」を機能語要素処理の段階で訳す。

Jaw では翻訳規則は表現構造と称するオブジェクトのメンバー変数の値という形に変換して表現する。図2はその様子を示している。

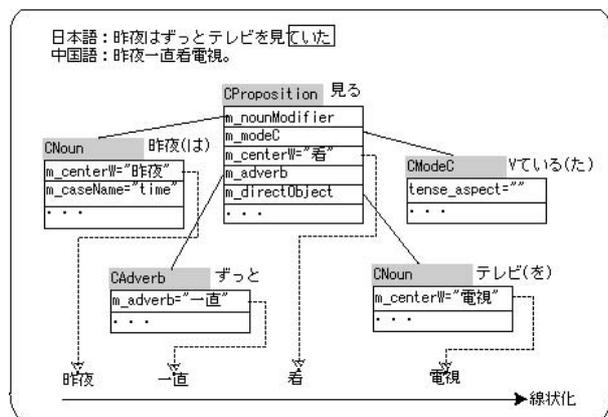


図3 - 「彼は怒っていた」に対する表現構造

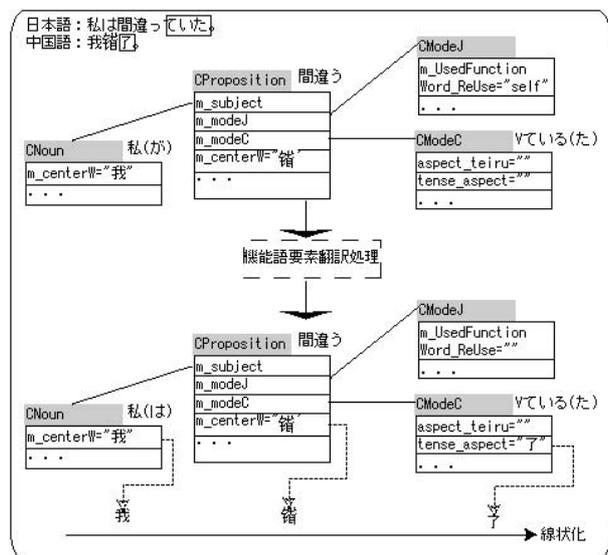


図4 - 「私は間違っていた」に対する表現構造

図3は「昨夜はずっとテレビを見ていた。」に対する中国語への翻訳の表現構造を示す。時点を表す単語「昨夜」がVに係るので規則①によって「Vている(た)」の訳語を無しにする。図4は「私が間違っていた。」に対する中国語への翻訳の表現構造を示す。この場合には規則②によって「Vている(た)」の訳語無しに訳す。更に図4の下方で示すように機能語要素翻訳処理の段階で機能語要素「た」を中国語の「了」に訳す。

### 3.2 Fw型パターン

Fw型パターンとなる機能語独立文節は「VUとすれば(とすると)」、「Qと(と)」などである。図5は「VUとすれば」に関するパターンとして、独立文節機能語「Vう」が字面で係るという形を示している。翻訳規則に関する処理は従来の通り[2]で行われる。

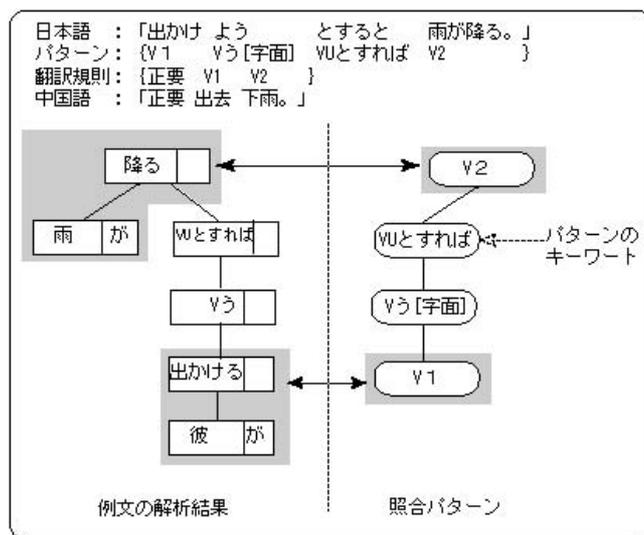


図5 - 「出かけよう」とすると雨がふる。」を処理するパターン

## 4 パターン照合処理に置ける問題点

### 4.1 接続詞を二つ持つパターン木に関するパターン組み立て直し処理

パターン組み立て直し処理とはパターン照合[3]の前に、入力文に応じて組立てたパターン木へ、関連パターンを加えて組み立て直すことである。接続詞を二つ持つパターン木に関するパターン組み立て直し処理については二つの場合がある(図6)。図中には次の三つのパターンがある。

パターンA: 既に組立てた接続詞を二つ持っているパターン木;

注: パターン木といっても、大きなパターンと認められる。

パターンB: 加えようとする述語のBase型パターン[3];

パターンC: 加えようとする機能語独立文節のFn型パターン;

場合①はパターンAとパターンBとの組み立て直しである。場合②はパターンAとパターンCとの組み立て直しである。黒影がある部分を見ると、パターンAの“接続詞2”はパターンB(或はパターンC)

の“述語1”に繋がれている。これを、場合①ではパターンBの根ノードに繋ぎ直す。場合②ではパターンCの葉ノードに繋ぎ直す。

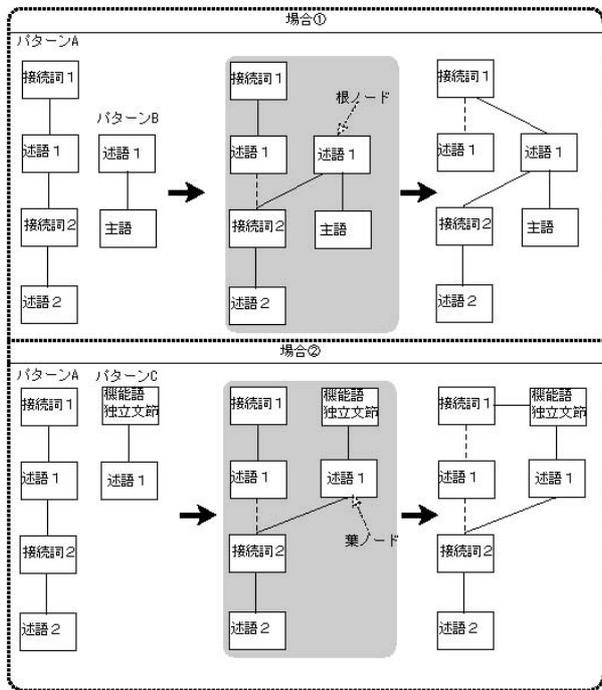


図6—パターン組み立ての二つの場合

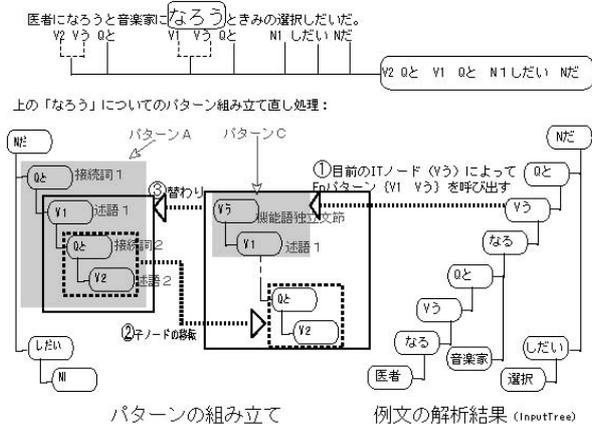


図7—図6の場合②の例

図7は図6の場合②の実例であり、次のパターンA'にパターンCを加えて組み立て直すことを三つのステップで示している。

パターンA': {V2 Qと V1 Qと N1しだい Nだ}

パターンC: {V1 Vう}

#### 4.2 機能語独立文節付き連体修飾文の照合

用言に関する連体修飾文のパターン照合[4]では自立語部分のパターンを変換することによって、係る

用言と受ける名詞との関係を示す。機能語独立文節があった場合、自立語部分と機能語独立文節部分に分けて別々パターン照合を行いし、その照合結果としての翻訳木 [TranslateTree] (TT) を再び合わせる。「川の両側に植えられている花。」を例文として詳しく述べる。

① 例文の解析結果(IT)から機能語独立文節部分を外して、その過程で機能語独立文節情報を収集する。外れた部分は末のノードに仮Fn型ノード「endfunc」をつけて“機能語IT”と呼ばれる。その「endfunc」は機能語ITの末ノードを示す。

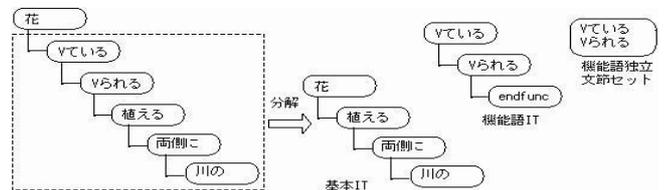


図8—機能独立文節付き連体修飾文のITの分解

② 機能語ITに対してパターン照合を行って、結果として翻訳木(TT)に変換させる。「endfunc」というパターン木[PatternTree](PT)を先に用意しておく。

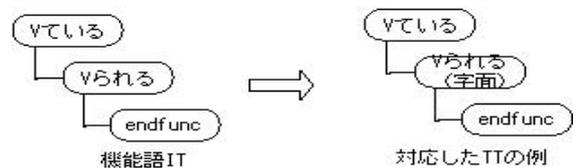


図9—連体修飾文の機能独立文節部分のパターン照合

③ 基本ITに対して、最初収集された機能語独立文節情報をパラメータとして連体修飾処理[4]を行って、結果として翻訳木(TT)に転換させる。

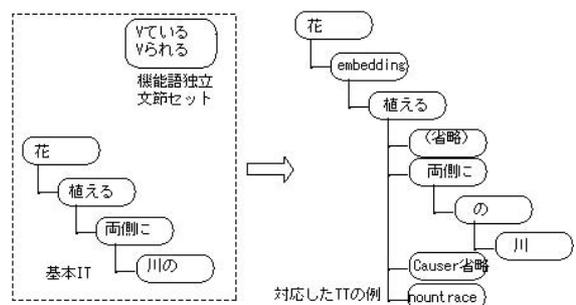


図10—機能語独立文節付き連体修飾文のパターン照合

④、②と③で生成した翻訳木(TT)を併合し、例文に

対応した翻訳木(TT)を生成する。

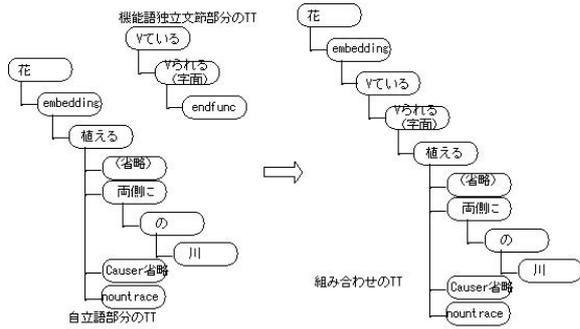


図 11—機能独立文節付き連体修飾文のパターン照合

### 4.3 機能語独立文節の字面パターン

「手をつけられない」、「あわをふかせる」、「期待に胸を膨らませる」など機能語独立文節つき慣用句について、機能語独立文節に関する従来の翻訳処理が適用できない。そのような慣用句について、機能語独立文節も字面として Base 型パターン翻訳規則を書く。詳しい翻訳処理は{期待に胸を膨らませる}について述べる。

例文「彼は期待に胸を膨らませる」の字面パターンでの照合

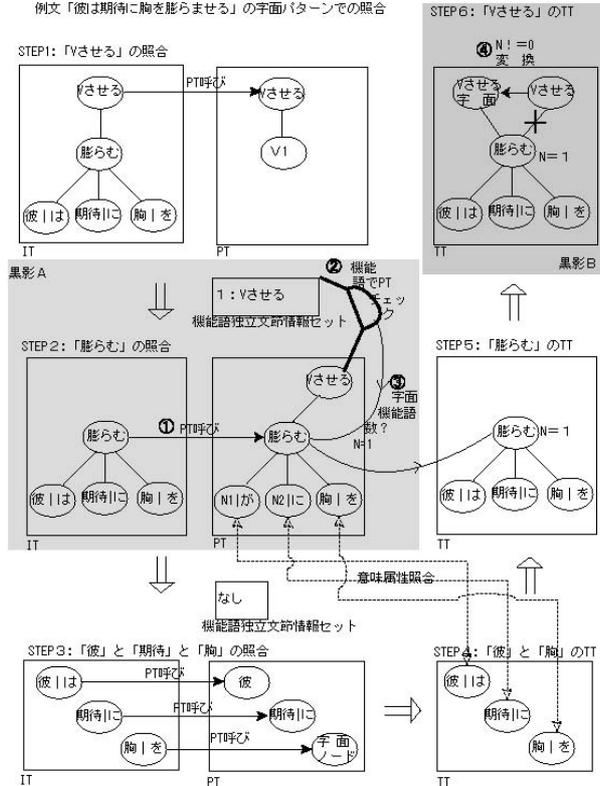


図 12—機能語独立文節の字面パターンの照合処理の流れ

「膨らむ」に関する字面パターン翻訳規則は次に示す：

パターン:<N1>が <N2>に 胸を(字面) 膨らむ Vさせる(字面)

翻訳規則:<N1> 充滿 <N2>

ステップ：

図 12 の黒影 A に示すように①と②、③を行う。

- ① 解析結果の IT ノード「膨らむ」によって「膨らむ」に関する上の字面パターンを呼び出す。
- ② 収集した機能語独立文節セットを用いて呼び出されたパターンが適切か否か判別する。非適切な場合、目前のパターンに関する処理を終了する。
- ③ 適切になった場合、そのパターンが持っている字面機能語独立文節の数 (N=1) を述語の TT ノード「膨らむ」に保つ。

図 12 の黒影 B に示すように、④を行う。

- ④ TT ノード「膨らむ」の上の N 個機能語独立文節を字面ノードに変換させる。

## 5 おわりに

機能語独立文節に関するパターン翻訳規則処理及びパターン照合処理における問題点への対応について述べた。機能語独立文節の翻訳処理について、今後翻訳規則を増やし、更に多くの翻訳実験で検証し、不足の点を更に洗い出す必要がある。

## 参考文献

- [1] 機能文節を導入した文節構造解析システム ibukiC(v0.20) について:池田尚志他:言語処理学会第14 回年次大会(2008)
- [2] 日本語から多言語への翻訳エンジン jaw, 宇野修一、福本真哉、田中伸明、松本忠博、池田尚志, 言語処理学会第 11 回年次大会, pp538-541, 2005.
- [3] 機械翻訳エンジン jaw における日本語パターンの記述形式及びパターン照合処理について:浅井良信, 穆貴彬, 玉置健二, 池田尚志:言語処理第13回年次大会 (2007)
- [4] 機械翻訳システム jaw における連体修飾構造の翻訳—日中, 日越を事例として— 浅井良信, 宇野修一, 田中友樹, ト朝暉, Nguyen My Chau, 池田尚志 言語処理学会第 12 回年次大会 発表論文集 p.284~287