

4N-5

融合方式で用いる  
機械翻訳用辞書について

糸井 理人、田中 穂積 (東工大 工学部)

1. はじめに

電子技術総合研究所で開発された融合方式による英日機械翻訳システム①は、解析、変換、生成の3過程を融合した翻訳を行なう。意味解析を行なう過程で単語の訳し分けを行ない、意味解析が終了し、意味構造が抽出されるとともに、翻訳結果がえられるという特徴がある。'TAKE'や'MAKE'についてはこの特徴を生かし、それぞれ20数通りの訳し分けができることが示されている。

このシステムでは、辞書の記述に意味表現用言語SRL②を用いている。しかし従来のSRLでは、ある単語の訳語を決定するために、文中において他の2つまでの語句しか参照できなかった。今回付加した新しい関数はネスト構造にすることができるため、任意個数の語句を参照して訳語を決定できる。また、冗長な括弧などを省いたこととあわせて、より見通しのよい辞書記述が可能になった。

本稿では、'TAKE'の辞書項目記述を例にこのシステムでもちいる機械翻訳用辞書について説明する。

2. 'TAKE'の辞書記述

図1に'TAKE'の辞書項目記述例をしめす。

図1(a)は、'TAKE'という見出しを持つ単語の品詞がV(動詞)であることを記述している。(b)は、'TAKE'がVT(他動詞)でありホーンビー動詞パターン③のVP6A, VP15A, ..., VP22を取り得ることを示している。これらは'TAKE'のメッセージとして、主として構文解析時に使う。

図1(c)以下は、意味解析及び翻訳のための記述である。この機械翻訳用辞書は、このようにシンタクスの情報とセマンティクスの情報をはっきり区別しているという特徴がある。

図1(c)は、'TAKE'のデフォルト訳語を、'ーを取る'にしている。

図1(e)...(g)は、目的語によって'TAKE'をどのように翻訳するかという記述である。(e)は目的語となる名詞の辞書項目に'TAKE'の訳語が記述されていることを表わす。例として'LIQUID'という名詞の辞書記述を図2に示す。図2(a)に、'HAVE'または

```
(TAKE V (a)
  ((VT VP6A VP15A VP15B VP2A VP12A (b)
    VP13A VP14 VP16B VP2A VP14 VP22) 0))
  ((TAKE 取 R . U) (c)
  unit
  (part-of)
  (self (a DEFAULT-CASE))
  (sf :+action)
  (subj HUMAN =agent) (d)
  (obj
    (((TO-FILL (@IDIOM1))) (e)
    (VEHICLE (f)
      (@HONYAKU * NI 乗 R . U)
      (@WAIT-AND-SEE (g)
        (with WEAPON (g1)
          (@HONYAKU2 goal1 WO *) (g2)
          (@HONYAKU1 = DE 乗っ取 R . U) (g3)
          (@WAIT-AND-SEE (g4)
            (goal T
              (@HONYAKU1 * 乗っ取って行 K . U)))
          =instrument
          goal1=obj)
            (goal T (g5)
              (@HONYAKU1 * 乗 TTEIK . U)))
          =goal1)
          .....中略.....
        ))
      (with (NOT ANIMAL) (h)
        (@HONYAKU = WOMOTTE *) (i)
      (to
        (((sf location)
          (@HONYAKU = HE *)
          (OBJECT
            (@HONYAKU = NOTOKOROBE *)))
          =goal)
          .....後略..... ))))
```

図1 TAKEの辞書項目記述

```
(LIQUID N ((UC) 0)
  ((LIQUID 液体)
  unit
  (part-of)
  (self (a SUBSTANCE))
  (sf)
  (idiom1 (OR HAVE TAKE) (a)
    (@HONYAKU * WO 飲 M . U)))
```

図2 LIQUIDの辞書項目記述

'TAKE'の目的語となっているとき、その動詞を'飲む'と訳せという記述がある。これにより'LIQUID'を上位概念にもつ'COFFEE'、'WINE'のような目的語を'TAKE'が持つときは、'飲む'と訳す。

上述のように、目的語の辞書項目に'TAKE'の訳語が記述されているときには、以下の(f)...(g)をおこなわない。(f)は、目的語が'VEHICLE'の時は'ーに乗る'と訳せ、という意味である。詳しく言うと、動詞'TAKE'のobjスロットの値として'VEHICLE'が当てはまるならば、次に来るLisp関数を実行する。関数 (@HONYAKU \* NI 乗 R . U) を実行すると、目的語の訳語はそのままにして('\*'は訳語

をそのまま当てはめるという意味)、それに助詞 'NI (に)' を付加する。そして最後に 'TAKE' のデフォルト訳語 '取る' を '乗る' に変える。その後、次の関数 @WAIT-AND-SEE を実行する。この関数は引数である (WITH ...) 以下をデモンとして保存しておき、他の処理が終わった後で実行する。最後の '=goal1' はスロット名を obj から goal1 へ変えることを意味する。

図 1 (h) は、'TAKE' が 'WITH' の率いる前置詞句を伴う時、'WITH' の目的語が 'ANIMAL' でなければその助詞を 'ーをもって' とすることを示す。そしてスロットの名称を with スロットとする。また (i) は、'TO' の率いる前置詞句を伴うとき、その目的語が場所を示す (その semantic feature が location である) ならば 'ーへ' と訳し、物体ならば 'ーのところへ' と訳すことを意味する。また '=goal' によってスロット名を goal に変える。

### 3. 関数 @WAIT-AND-SEE

例文 I take a plane with a gun to Moscow.  
(私はモスクワへ銃で飛行機を乗っ取って行く。)  
により、関数 @WAIT-AND-SEE について説明する。

'VEHICLE' は 'PLANE' の上位概念であるので、前述した図 1 (e) の処理を行なって、'TAKE' の goal1 スロットの値は 'PLANE' となる。この時は前置詞句のための処理 (図 1 (h), (i)) をまだ行っていないので、前置詞句に関する情報は得られない。そのため、'WITH'、'TO' などの前置詞がある場合に、どういう翻訳をするかというプログラムをデモンとして残すのが、関数 @WAIT-AND-SEE の働きである。目的語に続いて、前述のように前置詞の処理を行ない、'TAKE' の各スロットに次のような値が入る。

```
goal1 = ' PLANE    (飛行機に) '
with   = ' GUN     (銃をもって) '
goal   = ' MOSCOW  (モスクワへ) '
```

このように、単文の処理が一通り終わった段階では、それぞれの語句はすでにスロットの値となっているので、ここではじめて @WAIT-AND-SEE で作ったデモンを起動させる。@WAIT-AND-SEE の引数 (デモン) は (g1)... (g5) 及び (g6)... (g7) の 2 個である。(g1) は WITH スロットの値が 'WEAPON' のとき、以下の (g2)... (g5) を実行せよ、という意味である。例文では、WITH スロットの値は 'GUN' あるので、以下を実行する ('WEAPON' は 'GUN' の上位概念)。(g2) の

関数 @HONYAKUI では、'銃' に付いた助詞を 'で' に変え、'TAKE' の訳語を '乗っ取る' とする。(g3) の関数 @HONYAKUI2 では goal1 スロットの値 '飛行機に' の助詞を 'を' に変える。(g4) で再び @WAIT-AND-SEE を実行し、デモン (g5) をつくる。

ひとつの @WAIT-AND-SEE で作られるデモンは、COND 文のように択一選択で実行する。つまり上のように 1 つめのデモン (g1)... (g5) が実行されたときは、2 つめの (g6)... (g7) は実行しない。

@WAIT-AND-SEE が、(g) のようにネスト構造の時、内側で作られたデモン (g5) は、外側のデモン (g1)... (g5) の実行終了後すぐに起動される。(g5) では goal スロットが存在するとき ('T' は存在すればよいことを示す)、次の関数 @HONYAKUI を実行し、'乗っ取って行く' と訳す。

### 4. まとめ

@WAIT-AND-SEE を使ったこのような辞書記述は語イTRANSFUA用の規則がコンパイルされている。そして意味解析の過程において、その時点での最善の訳語を選択するとともに、さらにどのような語イTRANSFUA規則をもちいるかが絞られていく。また、一通りの処理が終わった段階でデモンを起動させるため、各スロットの値となる語句の順序に影響されないという特徴がある。

また、TO-FILL メソッド (図 1 (e)) によって、動詞の辞書記述だけが巨大化することを防ぐことができる。

今後抽出された意味構造を用いて、指示代名詞の参照先を決めるといった文脈処理を行ない、高度な機械翻訳システムの実現とともに、意味構造を用いて文理解の結果にたいする質問応答が可能になるようにしたい。それにより言語理解を通じた機械翻訳システム実現の可能性を示してみたい。

### 参考文献

- ① 田中穂積：「解析から合成までを融合した英日機械翻訳システム」、日経エレクトロニクス、pp. 275-293、1983. 8. 29
- ② 田中、元吉、安川：「意味表現用言語 S R L の機械翻訳への応用」、情報処理学会自然言語研究会資料、31-5、1982. 5
- ③ ホーンビー：「英語の型と語法」、オックスフォード大学出版局、1981.