

機械翻訳における辞書データベースの運用方式†

中 村 順 一^{††} 辻 井 潤 一^{††} 長 尾 真^{††}

機械翻訳システムで使用する辞書をどのようにして作成し、翻訳処理の際に使用するかは、翻訳システムの規模が大きくなるにつれ大きな問題となる。また、詳細な言語情報を記述した大規模な辞書は、機械翻訳に限らず、自然言語処理システムを研究したり、開発したりする際の貴重な資源であるので、これらを特定の機械翻訳システムにあまり依存しない形式で作成できれば、今後の自然言語処理の研究やシステム開発に非常に有効である。そこで、作成する辞書データベースと翻訳処理の際の辞書データとを明確に分離し、辞書データベースには、単語の言語的な性質を中立的に記述し、解析や生成といった翻訳の処理手順に依存した記述は、汎用辞書データベースから作成される翻訳処理用の辞書データのみに存在するようにした。これにより、辞書の作成と翻訳処理用の文法の作成が独立に行えるようになり、翻訳システムの性能を向上するための両者の修正や改良が、容易かつ精密に行えるようになった。本論文では、機械翻訳用の辞書データベースが満たすべき条件について考察し、それに基づき筆者らがを行っている日英・英日機械翻訳システムの開発 (Mu プロジェクト) において作成した、特定の翻訳システムとは独立な辞書データベースの構成と、実際の翻訳処理に使用するための各種処理用辞書を作成するためのソフトウェアについて述べる。

1. はじめに

機械翻訳システムにおいて、辞書をどのようにして作成し、機械翻訳用の文法により使用するかは、システムが大規模なものになるにつれ、大きな問題となってくる。辞書そのものは、自然言語処理システムを開発したり、研究する際の貴重な資源であるので、これを特定のシステムにあまり依存しない形式で作成することは、今後のこの分野の研究・開発にとって重要であると考えられる。本論文では、言語処理用の辞書データベースが満たすべき条件について考察し、それに基づき Mu プロジェクト¹⁾において作成した汎用辞書データベースの構成、その辞書データベースから Mu プロジェクトにおける日英/英日機械翻訳システムのための各種の辞書を作成するためのソフトウェアについて述べる。

2. 言語処理用の辞書データベース

言語処理用の辞書データベースを作成する場合、計算機可読の形になっている市販の辞書を活用する方法と²⁾⁻⁴⁾、開発しようとしている特定のシステムに合わせた専用の辞書を作成する方法とがある。市販の辞書は、特定の使用目的を想定していないので、多様な目的に使用できる可能性がある。しかし、本来人間が読むことを前提としており、記述の形式があまりにも自

由なために、その記述内容を計算機により活用するには、データ変換に多大な労力を要する。これに対して、特定の言語処理システム専用の辞書は、そのシステムにとって最適な形のもので作成できる。しかし、処理のための文法に対する独立性が低いことや、特定の処理システムだけに必要な情報が様々な個所に埋め込まれたり、柔軟性を求めるあまり、手続的な記述が辞書に埋め込まれたりする⁵⁾ため、他の用途に使用することは不可能になる。

両者とも長所とともに短所があるので、計算機で利用することを前提とした、両者の中間的な形式の汎用辞書を考えることにより、優れた辞書データベースが設計できる。そこで、以下の点を考慮して汎用辞書データベース・システムを設計、作成した。

(1) 辞書の作成、保守を翻訳処理システムとは独立に行えるようにし、ひとつの辞書をいろいろな用途に使用する。汎用辞書を作成する際には、それぞれの語の言語的な情報だけに注目し、宣言的に表現し、辞書を使用する際には、汎用辞書に対して形式変換を行う。これにより、汎用辞書の内容が分かりやすくなり、いろいろな用途、例えば、解析用と生成用に使用することが可能になる。

(2) 辞書データに構造を持った記述形式を許す。これにより、複雑な辞書情報が表現でき、かつ、プログラムにより容易に扱える。さらに、新たに辞書情報を追加しても、その影響が、個々の単語だけにとどまり、辞書データベース全体に広がらない。

(3) 単語固有の言語現象を十分詳しく記述できるようにする。これは、宣言的な形で行うことを許し、

† A Utilization Method of Dictionary Databases in Machine Translation by JUN-ICHI NAKAMURA, JUN-ICHI TSUJII and MAKOTO NAGAO (Department of Electrical Engineering, Kyoto University).

†† 京都大学工学部電気工学第二教室

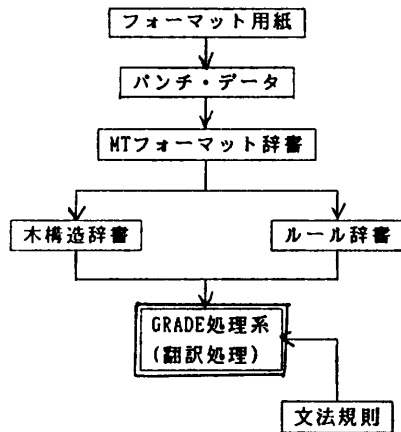


図 1 辞書の形式変換の流れ

Fig. 1 The flow of data transformations of the dictionaries.

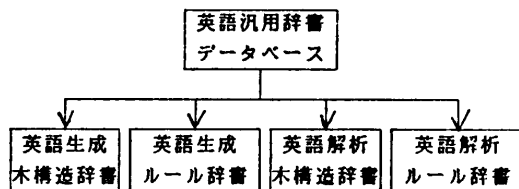


図 2 辞書データベースからの使用辞書の生成

Fig. 2 Transformation of the dictionaries for processings from the dictionary database.

そこから、処理アルゴリズムに合わせた書き替え規則の形に合成し、翻訳処理に使用できるようにする。

3. Mu プロジェクトにおける辞書データベース

Mu プロジェクトでは、日本語、日英変換、英日変換、英語の 4 種の辞書データベースを作成している。それぞれの辞書は、まず、辞書記述者が記入用の用紙（フォーマット用紙と呼ぶ）に個々の単語の情報を記入することにより作成される。図 1 に示すように、フォーマット用紙の記述は、パンチ入力され、MT フォーマット辞書という汎用辞書データベースに整理され、これを元にして翻訳システムで実際に使用する辞書が作られる⁶⁾。また、例えば、図 2 に示すように、英語の汎用辞書データベースからは、解析用の辞書と生成用の辞書という、全く使用目的の異なる辞書が生成されている。以下では、4 章と 5 章で汎用辞書データベース（MT フォーマット辞書）および翻訳処理用の辞書の内容について述べ、6 章で、辞書の形式変換の方法について述べる。

4. 汎用の辞書データベース

汎用の辞書は、まず、図 3 に示すような記入用のフォーマット用紙に所定の情報を書き込むことにより作成されるが、フォーマット用紙に記入される情報は、純粋に言語的なものであり、特定の処理システムには依存しない。したがって、フォーマット用紙に記入していく作業も、その語の純粋に言語的な性質だけに注目して記入していけば良く、処理システムの手続的な動作を全く知る必要はない。

フォーマット用紙に記入された言語情報は、パンチ入力され、図 4 に示す計算機可読な形式に変換される。この形式を MT フォーマット（以下 MTF と略す）と呼ぶ。図 3 と図 4 (a) から分かるように、フォーマット用紙の内容と MTF の内容とは等価なものであり、計算機による管理の対象となるのは、この MTF 辞書である。そこで、ここでは、以下の議論に必要な範囲で、MTF 辞書の内容について述べる（フォーマット用紙に記入する言語情報の詳細については、文献 7) と 8)）。

図 4 (a) は動詞「affect」の英語辞書の MTF の内容である。これには、

- (ア) 形態的な情報
 - 語尾変化 (INFLECTION) は規則変化 (REGULAR)
- (イ) 派生語の情報 (DERIVATION)
 - 動詞派生は effect
- (ウ) 格フレームの情報 (C-FRAME)
 - 表層格の主語 (SUBJ)
 - <-> 深層格の動作主体
(CPO, Causal POTency)
 - 表層格の直接目的語 (OBJ 1)
 - <-> 深層格の対象 (OBJ)

等の英語の動詞の言語的な情報が表現されている。

図 4 (b) は日本語の名詞「影響」の日英変換辞書の MTF の内容である。これには、

- (ア) 特に条件がない場合の一般的な訳語 (E-LEX)
 - 影響 -> effect
- (イ) 被修飾句による訳し分け (MODIFIED)
 - N への影響 -> effect on N
- (ウ) 共起する動詞による訳し分け (CORRESPONDENCE)
 - N1 が N2 に影響を与える

No.	16				
Lexical Unit	affect				
Sub-Category	VERB				
Component	Main Verb	affect			
	Adverbial Particle				
	Adverb-Correlative				
	Preposition				
	Prep-Correlative				
Subject Code					
Usage ID	1				
Semantic Category					
Verb Pattern	I / L / (T) / V / X / D				
Aspect	STATive / TRANSitive / (PROCEss) / COMPLETive / MOMentary				
Voice	++ / (+) / -	Subject in D-verb passive	1 / 2		
Volition	+ / (-)				
Agent of to-inf.	SUBJect / OBJect				
	Surface Case	Deep Case	Syntactic Form	Semantic Codes	
1	SUBJ	CPD			1
2	OBJ1	OBJ			1
3					
4					
Remarks					

図 3 英語動詞のフォーマット用紙
Fig. 3 Format form for English verb.

```

1 ((SEQ 16)
2 (E_LEX affect) ; 見出し語
3 (E_CAT V)
4 (INFLECTION REGULAR) ; (7) 活用型
5 (USAGE
6 ((E_UID 1)
7 (DERIVATION ; (i) 派生語の情報
8 (E_DERIV_STATE_ACTION effect)
9 (E_DERIV_STATE_ACTION_UID 1))
10 (COMPONENT (E_MAIN_VERB affect))
11 (E_VERB_PAT T)
12 (E_VERB_ASPECT PROC)
13 (E_VERB_VOICE /+)
14 (E_VERB_VOLITION /-))
15 (C-FRAME ; (9) 格フレームの情報
16 ((E_SURFACE_CASE1 SUBJ) ; 表層格
17 (E_DEEP_CASE1 CPO) ; 深層格
18 (E_OBLIGATORY1 1))
19 ((E_SURFACE_CASE2 OBJ1) ; 表層格
20 (E_DEEP_CASE1 OBJ) ; 深層格
21 (E_OBLIGATORY1 1))))))

```

(a) 英語の動詞「affect」の英語 MT フォーマット辞書

(a) MT format dictionary of the English verb "affect."

```

1 ((SEQ 6238)
2 (J_LEX 影響) ; 見出し語
3 (J_CAT 名詞)
4 (USAGE
5 (E_LEX effect) ; (7) 特に条件がない場合の
6 (E_UID 1) ; 一般的な訳語
7 (E_CAT N)
8 (MODIFIED ; (i) 被修飾句による訳し分け
9 ((J_BKK_LEX へ) ; N1への影響
10 (J_DEEP_CASE 受け手) ; -> effect on
11 (E_PREP_LEX on)
12 (E_LEX effect)
13 (E_CAT N)))
14 (CORRESPONDENCE ; (9) 動詞との共起による訳し分け
15 ((J_LEX 与える) ; N1がN2に影響を与える
16 (E_LEX affect) ; -> N1 affect N2
17 (E_CAT V)
18 (E_UID 1)
19 (CASE_FRAME ; (i) 動詞「与える」と「affect」の
20 ((J_SURFACE_CASE が) ; 格フレームの日英の対応関係
21 (J_DEEP_CASE 主体)
22 (E_SURFACE_CASE SUBJ)
23 (E_DEEP_CASE CPO))
24 ((J_SURFACE_CASE を)
25 (J_DEEP_CASE 対象)
26 (J_POS_FLAG DEL)) ; 「影響」が対象格
27 ((J_SURFACE_CASE に へ)
28 (J_DEEP_CASE 受け手)
29 (E_SURFACE_CASE OBJ1)
30 (E_DEEP_CASE OBJ1)))
31 ))))

```

(b) 日本語の名詞「影響」の日英変換 MT フォーマット辞書

(b) MT format dictionary of the Japanese noun "EIKYOU" from Japanese into English.

図 4 MT フォーマット辞書の例

Fig. 4 Examples of MT format dictionaries.

-> N1 affect N2

という日英の用法の対応関係が表現されている。

MTF 辞書は、情報の種類を表す属性名とその値の組みの形で表現されており、辞書情報の取り出しや、新たな情報の追加が容易に行える。また、一般のフレーム型の表現形式と同様に、ある属性名に対する属性値が再び属性名と値の組みで表現されることも許している。そこで、「ひとつの動詞には、複数の格フレームがあり、それぞれの格フレームごとに、属性と格フレームの内部構造が存在する」といった辞書記述が本来的に持つ構造を自然に表現することができる。

5. 翻訳処理に使われる辞書

MTF 辞書は、翻訳処理用の辞書とは独立なものである。そこで、図 1 に示したように、MTF 辞書から、文法規則が実際に使用する翻訳処理用の辞書を作成する。この形式変換の過程を設けることにより、汎用の辞書データベースとして管理・運用する MTF 辞書と、特定の翻訳処理システムに合わせた柔軟で効率の良い処理用辞書との独立性が得られることになる。

現在、使用している翻訳処理用の辞書には、木構造形式の辞書(1)とルール形式の辞書(2)がある。(1)は、文法規則が参照するための辞書で、宣言的な情報が表現されている。(2)は、文法処理を行っている過程において、適宜呼び出されて適用される単語個別の規則を記憶した辞書で、書き換え規則の形で表現されている。

5.1 木構造辞書

木構造辞書には、品詞や派生語情報など、一般の文法規則が参照する情報が MTF から抽出されている。一般の文法規則が参照する情報には、

- (1) MTF 辞書に直接記述されている情報
- (2) 直接的には記述されていないが、MTF 辞書の情報から合成できる情報

```

1 ((E_CAT (N))
2 (E_LEX (effect))          : 特に条件がない場合の訳語
3 (E_UID (1))
4 (J_DIC_VARIANT (T))      : 被修飾句による訳し分けフラグ
5 (J_DIC_DERIV (与える))   : 共起動詞による訳し分けフラグ
6 ))

```

図 5 名詞「影響」の日英変換用木構造辞書

Fig. 5 Tree-format dictionary for Japanese noun "EIKYOU" from Japanese into English, which is transformed from Fig. 4(b).

の二つがあり、木構造辞書には、これらの情報が文法記述用システム GRADE⁹⁾ の処理用の木構造の形式で蓄えられている。

木構造辞書の例として、名詞「影響」の日英変換用の辞書の内容を図 5 に示す。これには、図 4 (b) に示した MTF 辞書の内容に対応して、次に 3 種類の情報が蓄えられている。

(ア) 特に条件がない場合の一般的な訳語 (E-CAT, E-LEX, E-UID). 図 4 (b) の (ア) の部分の情報が図 5 の 1~3 行に変換される。これは、MTF 辞書に直接記述されている情報である。

(イ) 被修飾句に関する情報の存否を示すフラグ (J-DIC-VARIANT). 図 4 (b) の (イ) の部分の情報 (MODIFIED) が存在する場合、T になり (図 5・4 行目)、被修飾句が存在する場合に、この単語は特別な扱いをしなければならないことを示している。これは、MTF 辞書には、直接記述されていない情報である。

(ウ) 共起する動詞に関する情報の存否を示すフラグ (J-DIC-DERIV). 図 4 (b) の (ウ) の部分の情報 (CORRESPONDENCE) が存在する場合、共起する動詞の見出し語 (図 4 (a)・15 行目) が、図 5 の 5 行目のように、J-DIC-DERIV の値になる。これは、この名詞「影響」が「与える」に支配されている場合には、特別な扱いをしなければならないことを示している。これも、MTF 辞書には、直接記述されていない情報である。

このうち、(イ) と (ウ) の場合の具体的な処理内容はそれぞれ、この単語の日英変換用のルール辞書に文法規則の形で蓄えられている。J-DIC-VARIANT と J-DIC-DERIV は、共に、その単語が単語個別の特別な処理をするための辞書規則を持っているかどうかを一般の文法規則に対して示すものであり、一般の文法規則は、実行時にこれらの属性を検査し、ルール辞書に蓄えられている辞書規則を直接呼び出すことによって、この単語に固有な処理を行う。これらの属性は、

語の言語的性質を示しているというよりも、Mu プロジェクトで作成している翻訳システムという、特定の処理システムの手続上必要になるものである。そこで、MTF 辞書には記入せず、MTF 辞書から木構造辞書を作成する際に MTF 辞書中の属性と属性値から合成され、付加される。

5.2 ルール辞書

MTF 辞書には、4 章で述べたように、

- (1) 日英の動詞の格フレームの対応関係
- (2) 日本語や英語の格フレーム (表層格と深層格) の対応関係)

(3) 複数の語 (例えば、名詞と動詞) の共起関係等の単語個別で、複雑な構造を持った情報も宣言的に表現されている。これらの情報を複雑な構造を持ったまま木構造辞書に記憶しておき、一般的な規則で解釈しながら翻訳処理を行うよりも、あらかじめ、単語個別の文法規則 (木構造の変換規則) に変換しておき、処理の途中で必要に応じて直接起動した方がよい。そこで、ルール辞書には、複雑な構造を持つ言語情報を GRADE の文法規則の形で蓄えておく。

また、MTF 辞書には、単語個別の情報がどのような使われ方をするかとは無関係に宣言的な形で表現されているので、同一の記述をいろいろな用途のための書き替え規則に変換することもできる。例えば、英語の動詞の格フレームの記述は、

- (1) 英語生成用の文法規則
(深層格から表層格への写像)
- (2) 英語解析用の文法規則
(表層格から深層格への写像)

という、全く目的の異なった文法規則に変換されている。例として、図 4 (a) の英語の動詞「affect」から生成された解析用と生成用辞書規則とを図 6 に示す。

図 6 (a) は「affect」の英語解析用書き替え規則である。3~7 行が書き替え規則の適用条件を、8~13 行が適用結果を指定しており、図 7 に示す木構造の変換 (この場合は、節点に対する属性値の付加) を行うものである。この規則のうち、下線で示した部分が図 4 (a) の MTF 辞書から得られた情報で、その他の部分は、英語解析用書き替え規則全般に共通なものである。図 6 (b) は「affect」の英語生成用書き替え規則であり、図 7 の木構造の変換の逆を行う。MTF で同一の場所に記述された情報が、一方では規則の適用条件部に、他方では規則の適用結果を指定する部分に使用

```

1 affect_VG_ANALYSIS_V_1.RR; /* 英語解析用書き替え規則 */
2 MI: LEVEL(0,0); ORDER(1); TREE:

```

```

3 MC: %((SENT CASES));
4   SENT.E_UID = '1';
5   CASES: %(X1 VG X2); X1,X2:DISORDER_SKIP( %( X ) );
6   X1.E_SURFACE_CASE = 'SUBJ'; /* 表層格の条件 */
7   X2.E_SURFACE_CASE = 'OBJ1';

```

適用条件

```

8 CR: **;
9   X1.E_DEEP_CASE <= 'CPO'; /* 深層格の代入 */
10  X2.E_DEEP_CASE <= 'OBJ';
11  X1.E_GOVERNOR_ID <= VG.E_NODE_ID;
12  X2.E_GOVERNOR_ID <= VG.E_NODE_ID;
13  SENT.APPLIED <= 'T';

```

適用結果

```

14 END_RR.affect_VG_ANALYSIS_V_1;

```

(a) 英語の動詞「affect」の解析用辞書規則
(a) Dictionary rule of English analysis for verb "affect."

```

1 affect_V_GENER_1.RR; /* 英語生成用書き替え規則 */
2 MI: LEVEL(0,0); ORDER(1); TREE:

```

```

3 MC: %((V CASE));
4   CASE: %(X1 X2); X1,X2:DISORDER_SKIP( %( ? ) );
5   X1.E_DEEP_CASE = 'CPO'; /* 深層格の条件 */
6   X2.E_DEEP_CASE = 'OBJ';
7   V.E_UID = '1';

```

適用条件

```

8 CR: **;
9   X1.E_SURFACE_CASE <= 'SUBJ'; /* 表層格の代入 */
10  X2.E_SURFACE_CASE <= 'OBJ1';
11  V.E_MAIN_VERB <= 'affect';
12  V.E_VERB_TYPE <= 'VERBONLY';

```

適用結果

```

13 END_RR.affect_V_GENER_1;

```

(b) 英語の動詞「affect」の生成用辞書規則
(b) Dictionary rule of English generation for verb "affect."

図 6 MT フォーマット辞書から生成される辞書規則の例

Fig. 6 Examples of dictionary rules transformed from MT format dictionaries.

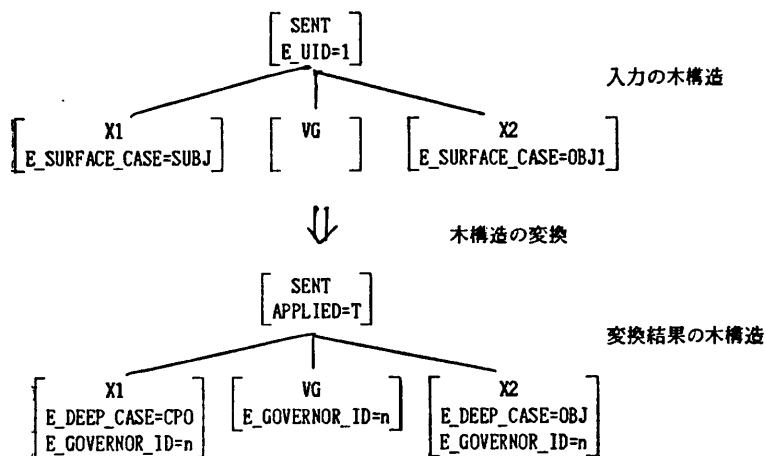


図 7 英語解析用辞書規則による木構造の変換

Fig. 7 Tree transformation by the dictionary rule for English analysis.

されていることが分かる。

ここに示した例は、比較的簡単な辞書規則であり、MTF 辞書に、より複雑な情報（例えば、特定の前置詞との共起、等）が記入されている場合は、書き替え規則ももっと複雑なものになる。

例えば、「insert」の英語動詞辞書には、前置詞「in」と共起することが記述されている¹⁰⁾。すなわち、

to insert a key in a lock

のように、前置詞「in」は、単純な場所を表すのではなく、深層格の「場所-終点」を示している、という情報が記述されている。この辞書記述は、英語生成用を使用する場合は、

日英変換結果中に、「場所-終点」格の名詞句があれば、それを支配する前置詞として「in」を選択する

という辞書規則に展開される。これに対して、英語解析用に使用する場合は、前置詞「in」の持つ係り受けのあいまいさを解消するための重要な手がかりであるので、

「in」が文中に表れた場合、「insert」が「場所-終点」格としてその前置詞句を支配する解釈を優先させる

という辞書規則に展開される。この場合、「場所-終点」格は、必須格ではなく、優先度が高い格であるとして扱う必要がある。

このように、同一の MTF の記述から、一方では、単純な前置詞の選択規則が、他方では、優先度を考慮した複雑な規則が生成される。このことは、汎用の辞書から、個々の翻訳システムに適した処理専用の辞書を作成することが可能であることを示している。

6. 形式変換のためのソフトウェア・ツール

これまでに述べたように、汎用の

MTF 辞書を翻訳処理に適した処理用の辞書に変換して使用する方式は、処理用の辞書を直接作成する一般的な方法では困難な、辞書記述の宣言性と翻訳処理の柔軟性とを両立させることを可能にした。しかしながら、この方式を実現するために必要な辞書の形式変換、すなわち、宣言的に記述されている MTF 辞書を、文法規則の形で記述されているルール辞書に変換することは、それほど容易ではない。MTF から文法規則の表現を生成するためのプログラムを

直接 LISP 等の汎用のプログラム言語で作成することも可能であるが、この場合には、MTF 辞書の形式そのものの変更や処理用の辞書の形式が変わった場合に面倒なことになる。そこで、こういった将来の変更や拡張に備えるために、形式変換を容易に記述できるソフトウェア・ツールが必要になる。

辞書の形式変換は、次の 2 段階に分離して考えることができる。

- (1) MTF 辞書から必要な情報を取り出す操作。
- (2) (1)で取り出された情報を実際の翻訳処理用の記述（例えば、文法規則）に展開する操作。

形式変換をこのように 2 段階に分けて行えば、MTF 辞書の仕様が変更された場合は、(1)の操作を、処理用の辞書の仕様が変更された場合は、(2)の操作を変更するだけで対応できる。これにより MTF 辞書と処理用の辞書との独立性をさらに高めることが可能になる。

そこで、(1)の操作を行うためにデータ変換ユーティリティを、(2)の操作を行うためにマクロ言語を、形式変換のためのソフトウェア・ツールとして作成した。このツールにより、MTF 辞書は、図 8 に示すように、まず、中間的な辞書の表現である「マクロを用いた辞書表現」(図 9)に変換される。この表現は、MTF 辞書から、文法規則を合成するために必要な情報だけ（例えば、SUBJ や CPO 等）を抽出したものである。次に、この中間的な辞書表現が、最終的な形

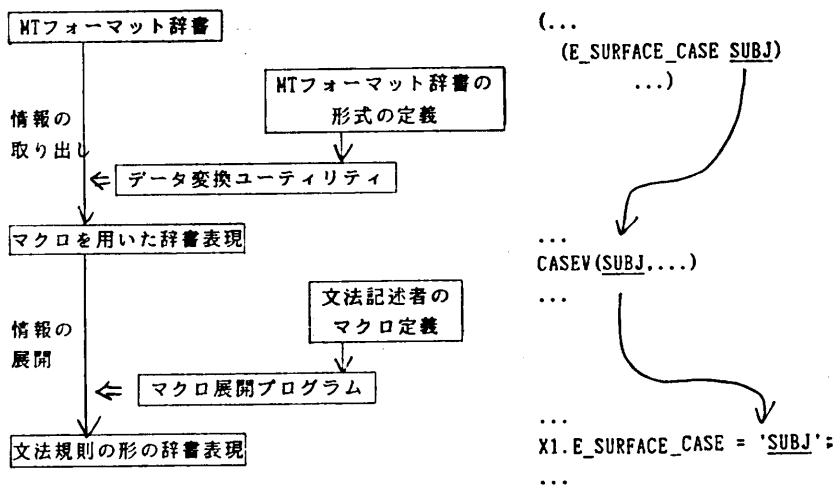


図 8 ルール辞書生成の流れ

Fig. 8 The flow of transformation of dictionary rules.

```
1 ANARV(affect,1,1,          /*書き替え規則の名前の情報*/
2 CASEV(SUBJ,CPO,OBJ1,OBJ)) /*格フレームの情報*/
```

図 9 マクロを用いた辞書表現（動詞「affect」の英語解析用規則）の例
Fig. 9 An example of dictionary format using macro-expression (English analysis rule for verb "affect").

式変換結果である「文法規則」の形に変換される。以下では、動詞「affect」の格フレームを英語解析用辞書規則に変換する過程を例にして、開発したソフトウェア・ツールについて述べる。

6.1 データ変換ユーティリティ

データ変換ユーティリティは、

- (1) MTF（属性名と属性値の組の形のデータ）中に表れる属性名とその属性値に対する操作（LISP プログラム）の定義（DEF-MTF-CONV）
- (2) 定義に従って LISP プログラムを実行する関数（MTF-CONV）

から構成されている¹⁰⁾。MTF 辞書からマクロを用いた辞書表現を作成するプログラムは、この二つを利用することにより、簡単に作成することができる。すなわち、(1)で MTF 辞書から必要な情報を取り出す方法を定義し、(2)の関数 MTF-CONV を呼び出すことにより、(1)の定義に従って MTF から情報を取り出して、マクロを用いた辞書表現を合成してゆくプログラムを作成すればよい。このように、MTF 辞書の形式の定義を行う(1)の部分とその定義を用いたプログラムとが独立に作成できるので、辞書の形式や内容が変更された場合にも対応が容易である。

```

1 (DEF-MTF-CONV ANAV-USAGE NIL NIL NIL      ; 定義の名前がANAV-USAGE
2   ((E_UID (SETQ IE_UID ?V) NIL)           ; 属性名とそれに対する操作の
3   (C-FRAME (SETQ !C-FRAME ?VS) NIL)       ; 定義
4   ))

```

(a) MT フォーマットからのデータの取り出し方の定義
(a) Definition of the data extraction from MT format.

```

1 (DEFUN WRITE-ANARV (E_LEX USAGE)          ; 書き替え規則を生成するマクロ
2   (LET ((CNT 1) (IE_UID "") (IC-FRAME NIL))
3     (LOOP (UNLESS USAGE (EXIT))
4       (MTF-CONV (POP USAGE) 'ANAV-USAGE) ; 必要なデータの取り出し
5       (FORMAT "ANARV(/C,/C,/C,/N" E_LEX CNT E_UID)
6       (WRITE-CASEGV !C-FRAME)             ; 格フレーム部の生成
7       (INCR CNT 1))))

```

(b) 生成用の LISP プログラム

(b) LISP program for the rule generation.

図 10 マクロを用いた文法記述を MT フォーマット辞書から生成する LISP プログラムの一部

Fig. 10 LISP program for generating macro-expressions from MT format dictionary (part).

英語動詞の解析用の「マクロを用いた辞書表現」を合成するための定義とプログラムの一部を図 10 に示す。定義は、『(〈属性名〉〈Lisp プログラム〉…)』の形式で行うことができる(図 10(a)の 2~3 行)。この例の場合は、属性名が E-UID と C-FRAME のデータに対して操作を行うことが指定されている。MTF 辞書中に指定した属性名のデータが存在した場合は、ユーティリティ側であらかじめ定義されている大域変数(?V や ?VS 等)に属性値が代入された状態で Lisp プログラムが実行されるので、この部分に必要な処理を定義しておけばよい。この例では、MTF 辞書から、E-UID と C-FRAME の値を取り出し、大域変数!E-UID に用例番号(USAGE-ID)を、!C-FRAME にその用例の格フレームの情報を代入する(SETQ)ことが示されている。

図 10(b)のマクロを用いた辞書表現を合成するプログラムは、関数 MTF-CONV により図 10(a)で定義し ANAV-USAGE を呼び出し(図 10(b)・4 行目)、必要なデータを MTF 辞書の一部(図 4(b)の 6~21 行の USAGE の部分)から取り出している。すなわち、関数 MTF-CONV を実行することにより、MTF 辞書の USAGE の部分から、E-UID の値と C-FRAME の値(図 4(b)の(ウ)の部分)とが取り出され、変数 !E-UID と !C-FRAME に代入される。そこで、5 行の関数 FORMAT により、E-LEX(見出し語)、USAGE の通し番号(CNT)、E-UID が、書き出される。次に、別の合成用のプログラム WRITE-CASEV に C-FRAME の値を渡すことにより、格フ

レーム部の情報が書き出される。関数 WRITE-CASEV でも、これと同様に DEF-MTF-CONV で定義されたデータの取り出し方を使用して、変換に必要な情報の取り出しと書き出しの操作が行われる。

このプログラムを用いて、図 4(b)の MTF 辞書から合成された、マクロを用いた辞書表現が図 9 に示したものである。1 行目が図 4(b)の 4 行目から、2 行目が、関数 WRITE-CASEV により合成される。ANARV と CASEV は、後で述べるマクロ定義の名前であり、SUBJ や CPO 等が図 3(b)の(ウ)の部分から取り出されたものである。

6.2 マクロ言語

マクロ展開プログラムは、マクロ定義に従って、文字列の置き換えを行うものである。マクロを用いた辞書表現は、文法記述者が作成したマクロ定義に従って GRADE の文法規則に展開される。文法記述者は、図 10 のような、MTF 辞書から抽出されたデータを、あらかじめ用意した文法規則の記述のテンプレートの適切な場所に埋め込む形式のマクロ定義を行えばよい。

マクロ定義の記述形式を図 11 に示す¹⁰⁾。例えば、

```
DEFINE (PAT (X,Y), MC; % (X Y);)
```

のようにマクロ PAT を定義すると、マクロ展開プログラムに対する入力(図 9 のマクロを用いた辞書表現)中に、『PAT (ADJ, NP)』という文字列があれば、これが『MC; % (ADJ NP)』という文字列に置き換えられる。また、単純な文字列の置き換えだけでなく、条件付きの置き換えも可能であり、文法記述者は必要に応じて複雑なマクロ展開を自由に行える。

英語生成用の動詞の文法規則を生成するマクロ定義

```
DEFINE(<文字列0>(<仮引数0>,<仮引数1>,...),
```

```
<文字列1>)
```

<文字列0>を<文字列1>で置き換える。仮引数がある場合は、その文字列も置き換えの対象になる。

```
IFELSE(<文字列0>,<文字列1>,
```

```
<文字列2>,<文字列3>)
```

<文字列0>と<文字列1>が等しい場合はこの形式全体を<文字列2>に、そうでない場合は<文字列3>に置き換える。

図 11 マクロ言語の記述形式 (一部)

Fig. 11 Format of the macro-language (part).


```

1  DEFINE(CASEV                                /* 英語解析用マクロ名 */
2      (IESC1,!EDC1,                            /* 第一格のパラメタ(表層格と深層格)
3          IESC2,!EDC2,)),                      /* 第二格のパラメタ(表層格と深層格)

4  MC: %((SENT CASES));
5      SENT.E_UID = '!EUID';
6      CASES: %(X1 VG X2); X1,X2:DISORDER_SKIP( %( X ) );
7      X1.E_SURFACE_CASE = '!ESC1';           /* 表層格の条件 */
8      X2.E_SURFACE_CASE = '!ESC2';

9  CR: **;
10     X1.E_DEEP_CASE <= '!EDC1';             /* 深層格の代入 */
11     X2.E_DEEP_CASE <= '!EDC2';
12     X1.E_GOVERNOR_ID <= VG.E_NODE_ID;
13     X2.E_GOVERNOR_ID <= VG.E_NODE_ID;
14     SENT.APPLIED <= 'T';
15 )

```

図 12 マクロ定義の例 (英語解析用の動詞の文法規則を生成するためのもの)

Fig. 12 An example of the macro-definition for generating English analysis rule for verbs.

の一部を図 12 に示す。この例の CASEV は、表層格を深層格に変換する書き替え規則の適用条件と適用結果の部分を生成するマクロ定義である。この定義を用いて図 10 の表現は図 6 (a) に示した GRADE の文法規則の書き替え規則に展開される (下線で示した部分が置き換えられる)。この例は、実際に Mu プロジェクトで使用されているマクロ定義を説明のために簡単にしたものである。支配する格の個数が違う場合や、5 章で述べたように、MTF 辞書にはもっと複雑な言語情報が記入されている場合があるので、実際のマクロ定義では、条件付きの置き換えも多用されている。

このように、文法記述者は、自分の文法に合わせてマクロ定義を行うことにより、自由に MTF に記述されている情報を活用することができる。また、将来、文法規則の形式を変更した場合にも、このマクロ定義を変更するだけで、これに対処することができる。

7. おわりに

Mu プロジェクトにおける辞書の運用方式について述べた。処理手順や文法処理に依存する記述と語の言語的な性質の記述とを区別し、辞書を 2 段階で管理する方式を採用した。この方式により、

- (1) 辞書が処理システムとは、独立に作成できるので、辞書記述の修正や追加が容易に行えるようになった。
- (2) 翻訳処理の柔軟さを低下させることなく、辞書中に言語情報を構造を持った形で宣言的に

表現することが可能になった。また、このため、辞書エディタの作成や、辞書内容の形式の検査用のプログラムの作成等が容易になった¹¹⁾。

- (3) 同一の辞書データを文解析と文生成といった異なった目的の翻訳処理に使用することが可能になり、辞書の汎用性が向上した。

という利点が得られた。

この方式を実現するために、辞書形式の変換用のソフトウェア・ツールとして、データ変換ユーティリティとマクロ言語を開発した。このツールにより、

- (1) 辞書から処理用の文法が必要とする記述を容易に生成できるようになった。
- (2) 辞書形式が変更された場合は、データ変換ユーティリティを利用したプログラムを、文法が変更された場合は、マクロ言語による定義を変更するだけで、各種の変更に柔軟に対処できるようになった。

という利点が得られた。

今後は、辞書中の単語の相互の参照関係を検査し、辞書全体の整合性を確かめる、といった辞書の保守のためのソフトウェアを開発すること等により、より有効な辞書データベースを構築していくことが必要である。また、各種のソフトウェアを作成する上でも、辞書データベースが人間にとって分かりやすい MT フォーマットの形式で蓄積、管理されていることは重要であると思われる。

謝辞 最後に、この辞書データベースの作成は、Mu プロジェクトの辞書作業グループ、および、処理作業のメンバの多大な労力の結果であることを記して、感謝の意を表する。なお、本研究は国の科学技術振興調整費による「日英科学技術文献の速報システムに関する研究」の一部として行ったものである。

参 考 文 献

- 1) 長尾 真: 科技厅機械翻訳プロジェクトの概要, 情報処理学会自然言語研究会資料, 38-2 (1983).
- 2) 鶴丸弘昭ほか: 国語辞典からの情報抽出とその構造化, 情報処理学会自然言語研究会資料, 43-6 (1984).
- 3) Nagao, M. et al.: An Attempt to Computer-

ized Dictionary Data Base, COLING 80, pp. 534-542 (1980).

- 4) 長尾 真ほか: 機械翻訳に対するロングマン辞書データベースの応用, 情報処理学会自然言語研究会資料, 29-5 (1982).
- 5) 井佐原均ほか: 融合方式に基づく機械翻訳, 情報処理学会自然言語シンポジウム予稿集 (1984).
- 6) 電子技術総合研究所ほか: 日英科学技術文献の速報システムに関する研究: 言語処理システムの開発に関する報告書 (1984).
- 7) 日本科学技術情報センターほか: 日英科学技術文献の速報システムに関する研究: 日-英科学技術用語辞書データベースの開発に関する報告書 (1984).
- 8) 長尾 真ほか: 機械翻訳における訳語選択と構造変換過程, 情報処理, Vol. 26, No. 11, pp. 1261-1270 (1985).
- 9) 中村順一: 文法記述用ソフトウェア GRADE, 情報処理学会自然言語研究会資料, 38-4 (1983).
- 10) 中村順一ほか: Mu プロジェクトにおける辞書の運用方式—日英変換辞書と英語生成辞書—, 情報処理学会自然言語シンポジウム予稿集 (1984).
- 11) 片桐恭弘ほか: Mu プロジェクトにおける翻訳支援環境, 情報処理学会自然言語研究会資料, 47-9 (1985).

(昭和 60 年 7 月 10 日受付)

(昭和 61 年 6 月 19 日採録)



中村 順一 (正会員)

1956 年生. 1979 年京都大学工学部卒業. 1981 年同大学院修士課程修了. 1982 年同大学院博士課程中途退学. 1982 年 11 月より, 京都大学工学部電気工学第二教室助手. 自然言語処理, 機械翻訳, 人工知能の研究に従事. 訳書「人工知能入門」(分担), 「人工知能ハンドブック」(分担). 電子通信学会, 日本ソフトウェア科学会各会員.



辻井 潤一 (正会員)

昭和 24 年生. 昭和 46 年京都大学工学部電子工学科卒業. 昭和 48 年同大学大学院工学研究科修士課程修了. 同年京都大学工学部電気工学第二教室助手. 現在, 同教室助教授. 工学博士. 自然言語処理, 機械翻訳, 人工知能の研究に興味を持つ. 電子通信学会などの会員.



長尾 真 (正会員)

1936 年生. 1961 年京都大学工学部電子工学科修士課程修了. 京都大学工学部助手, 助教授を経て, 1973 年京都大学工学部教授, 現在に至る. 国立民族学博物館併任教授. 自然言語の機械処理, 機械翻訳, パターン認識, 画像処理などの研究を行ってきた.