

日英双方向機械翻訳システム BIJET の日本語処理と 翻訳実験について

吉武, 春光
九州大学大学院総合理工学研究科情報システム学専攻

三池, 誠司
株式会社東芝 | 九州大学大学院総合理工学研究科情報システム学専攻

田町, 常夫
九州大学大学院総合理工学研究科情報システム学専攻

<https://doi.org/10.15017/17617>

出版情報 : 九州大学大学院総合理工学報告. 6 (2), pp.275-283, 1985-01-01. 九州大学大学院総合理工学研究科
バージョン :
権利関係 :

日英双方向機械翻訳システム BIJET の 日本語処理と翻訳実験について

吉武春光*・三池誠司**・田町常夫***

(昭和59年9月29日 受理)

Bidirectional Japanese English machine translation system BIJET —The processing part of Japanese and translation experiments—

Harumitsu YOSHITAKE, Seiji MIIKE and Tuneo TAMATI

The authors have recently constructed an experimental machine translation system named BIJET (Bidirectional Japanese English Translation system). This system consists of three parts: a Japanese processor, an English processor and a transfer processor. Both Japanese and English processors are divided into a morphological analyser, a morphological synthesizer and a syntactic processor. A transfer processor is divided into a Japanese-English transfer processor and an English-Japanese one. This system has the following features.

(1) BIJET works in two modes: translation from Japanese to English and from English to Japanese. The Japanese syntactic processor works as analyser in Japanese-English translation mode and as a synthesizer in English-Japanese mode. The English syntactic processor works in the same way.

(2) This system uses transfer method. Its intermediate representations are English D-tree and Japanese D-tree. Japanese D-tree and linguistic information in Japanese are newly proposed. Japanese D-tree is constructed from dependency relation and RENSETSU rule. In transfer processors, conversions, "transfer", are made between Japanese and English D-free.

This paper gives the details of the Japanese processor and the transfer processor in BIJET, then discusses translation experiments and its results.

1. 緒 言

筆者らは日英双方向機械翻訳システム BIJET (Bidirectional Japanese English Translation system) の開発を行なった。BIJET は主に Prolog によって記述されたトランスファ方式による機械翻訳システムであり、日→英翻訳と英→日翻訳という2つの動作モードを持っている。前稿¹⁾において BIJET の概要と英語処理部について述べたが、本稿では日本語処理部、トランスファ部と全体の翻訳実験について述

べる。

2. 日本語 D-tree モデルと日本語依存文法

本章では翻訳における日本語側の中間表現である日本語 D-tree とそれを生成する依存方法について述べる。

2.1. 日本語 D-tree モデル

本モデルは、接続規則²⁾³⁾によって組み立てた文節を最小構文単位として依存構造を組み立てている。

(JDM1) 機械が識別すべき最小の文要素を語とする。

(JDM2) 最小構文単位を文節とする。

(JDM3) 文の構文構造を D-tree で示す。これは

* 情報システム学専攻博士課程

** 情報システム学専攻修士課程 (現在(株)東芝)

*** 情報システム学専攻

機能要素を節点 (node), 述語を根 (root), 機能要素間の依存関係を枝 (branch) とする。

2.2. 日本語依存文法

上記の D-tree を生成する依存文法 G_J を形式的に次のように定義する。

$$G_J = (V_N, V_T, R, C, A, P, S)$$

V_N : 非終端記号の有限集合

但し $V_N = V_H \cup V_K \cup V_C \cup V_F$

ここで V_H : 品詞の有限集合

V_K : 機能要素の有限集合

V_C : 活用類の有限集合

V_F : 活用形の有限集合

V_T : 終端記号 (単語) の有限集合

R : 依存規則の有限集合

$$R = \{X_i(X_{j1}, X_{j2}, \dots, X_{jn}, *) | X_k \in V_N\}$$

ここで X_j : governor

X_{j1}, \dots, X_{jn} : dependent

*: governor の位置

カッコ内の * 以外の要素の順序は任意である。

A : 単語代入規則の有限集合

$$A = \{X \rightarrow x | X = (H_{i-1}, C_{i-1}, F_{i-1}, H_i, C_i, F_i), \\ H_i \in V_H, C_i \in V_C, F_i \in V_F, x \in V_T\}$$

なお, H_i, C_i, F_i はその単語のものであり, $H_{i-1},$

C_{i-1}, F_{i-1} はその単語の一つ前の単語のものを表す。

つまりその単語が接続できるのは $H_{i-1}, C_{i-1}, F_{i-1}$ をもつ単語に限るということを表している。

C : 接続規則の有限集合

$$C = \{X \cdot Y | X = (H_{i-1}, C_{i-1}, F_{i-1}, H_i, C_i, F_i), \\ Y = (H_i, C_i, F_i, H_{i+1}, C_{i+1}, F_{i+1}), \\ H_i \in V_H, C_i \in V_C, F_i \in V_F\}$$

なお, “.” は接続を表す。つまり二つの単語が接続できることを表している。

P : 機能要素割当規則の有限集合

$$P = \{Y \rightarrow y | Y \in V_K, y \in V_H\}$$

S : 初期記号の有限集合

ただし, $S \in V_K$

この定義に従う日本語依存文法 G_J が生成する D-tree を日本語 D-tree と呼ぶ。

2.3. 日本語の言語情報

日本語における品詞と単語情報の分類を Table 1 と Table 2 に示す。品詞の分類においては、用言の

Table 1 Parts of speech in Japanese.

Symbol	A part of speech
DET	連体詞
PRON	代名詞
NOUN	名詞
ADV	副詞
AUX	助動詞
VERB	動詞
	(断定の助動詞
	使役の助動詞
ADJ	(形容詞
	形容動詞
CONJ	格助詞「と」
COJ1	(従属の接続詞
	接続助詞
COJ2	(等位の接続詞
	接続助詞
PPRE	連用修飾の助詞
NPRE	連体修飾の助詞「の」
JSUB	(格助詞「が」「の」
	副助詞「は」
JDO	格助詞「を」
JIO	格助詞「に」
KOTO	形式
COMA	読点

Table 2 Word information in Japanese.

Symbol	Word information
TAN	(接尾語なし
	接尾語「さん」等
FUKU	(接尾語
	「たち」「ら」「方」等
1	自称
2	対称
3	他称
GENZAI	現在
KAKO	過去 (助動詞「た」)
TEINEI	丁寧
KANRYO	完了
HITEI	打消
ISHI	推量・意志
KYOKA	許可
SHINKOU	動作の存続
KINSHI	禁止
UKEMI	受身
KANOU	可能
KIBOU	希望
SUITEI	推定
DENBUN	様態・伝聞
GIMON	疑問
KINSHI	禁止

うち、主として動作・作用を表わす動詞を VERB とし、性質・状態を表わす形容詞及び形容動詞を共に ADJ として分類した。又、助詞が文の構造決定に有力な情報となることから、格助詞、接続助詞及び副助詞を CONJ, PPRE 又は NPRE という品詞に分類した。更に、助詞「が」「は」「の」「を」及び「に」は、特に別の品詞 JSUB, JDO, JIO と分類した。これ

Table 3 Relations between JYOSHI and function in Japanese.

JYOSHI	Function	Example
が は の	SUB (主語)	私が書く。 私は書く。 それは <u>私</u> 書いた本です。
に	IO (間接目的語)	私は <u>彼</u> に本を与える。
を	DO (直接目的語)	私は <u>彼</u> に本 <u>を</u> 与える。

は、これらの助詞が、それらを含む文節の機能要素を決定するからである。しかしながら、助詞と機能要素の結びつきは一意ではないので、厳密には構文情報のみでなく意味情報も取り入れて、文の機能要素を決定する必要がある。しかし、本研究では、翻訳対象文を中学校英語教科書の和訳文としたことにより、入力日本語文に **Table 3** の助詞と機能要素の結びつきが成立すると仮定した。

次に日本語における機能要素と述語機能パターンの分類をそれぞれ **Table 4~7** に示す。なお、述語機能パターンのうち、下線を引いた機能要素の並び及び述語 P の位置以外では、機能要素の並びは任意である。これは、日本語の性質として次のものを仮定したからである。

- (1) 述語は文末に位置する。

Table 4 Obligatory function.

Symbol	Function
SUB	主語
P	述語
PIV	叙述補助語
IO	間接目的語 (作用における動作の担い手など)
DO	直接目的語 (動作・作用の対象, 目的物を示す)
AC	副詞的補語 (動作・作用の方向, 起点や帰着する所を示す)
NO	目的語文 (引用文など)
NC	補語文 (動作・作用の対象としての動作・作用)
CM	読点 (、)

Table 5 Optional function.

Symbol	Function
SC	従属の接続詞・接続助詞
CC	等位の接続詞・接続助詞
S-NM	名詞修飾文
P-PM	述語修飾句
S-PM	述語修飾文

- (2) 被修飾語は修飾語の後に位置する。
- (3) 上記 (1), (2) が成立する限り、語順は任意である。

日本語の動詞、助動詞の活用を **Table 8** の活用辞書に示す。縦軸は活用類、横軸は活用形を表わす。活用形の用法を **Table 9** に示す。この活用辞書は語の

Table 6 Verb pattern type P1.

Pattern No.	Verb pattern	Example
P1-1	SUB+PIV+P	それは本です。
P1-2	SUB+P	(空が青い。 鳥が飛ぶ。
P1-3	SUB+DO+P	私は本を読む。
P1-4	SUB+IO+DO+P	私は <u>彼</u> に本 <u>を</u> やる。
P1-5	SUB+AC+P	私は <u>中国</u> へ行く。
P1-6	SUB+DO+PIV+P	私は <u>それを</u> 美しい <u>と</u> 思う。

Table 7 Verb pattern type P2.

Pattern No.	Verb pattern	Example
P2-1	SUB+NC+P	私は泳ぐのをやめる。
P2-2	SUB+IO+NC+P	私は <u>彼</u> に仕事をさせる。
P2-3	SUB+NO+P	私は <u>彼</u> が泳ぐのをみた。
P2-4	SUB+IO+NO+P	私は <u>母</u> に <u>弟</u> が明日帰ると言った。

Table 8 A part of conjugation dictionary

FORM TYPE	MI. NAI	MI. U	MI. YOU	MI. SERU	MI. ZU	YOU. MASU	YOU. TA	YOU. DA	YOU. NARU	YOU. NAI	SHU	TAI	KA	MEI
KA5	か	こ	X	X	X	き	い	X	X	X	く	く	け	け
GA5	が	ご	X	X	X	ぎ	X	い	X	X	ぐる	ぐる	げ	げ
1DAN		X		X	X			X	X	X			れ	ろ
KAHEN	こ	X	こ	X	X	き	き	X	X	X	くる	くる	くれ	こい
DOU	X	だろ	X	X	X	X	だっ	X	に	で	だ	ない	なら	X
KEI	X	かろ	X	X	X	X	かっ	X	X	X	く	い	けれ	X
DESU	X	X	しょ	X	X	X	し	X	X	X	す	す	X	X
TA	X	たろ	X	X	X	X	X	X	X	X	た	た	たら	X
DA	X	だろ	X	X	X	X	X	X	X	X	た	た	だら	X

X indicates that conjugation does not occur.

Table 9 How to use of conjection.

Conjugation form	How to use
MI, NAI	未然形で, ナイに付く
MI, U	ウ "
MI, YOU	ヨウ "
MI, SERU	セル "
MI, ZU	ズ "
YOU, MASU	連用形で, マスに付く
YOU, TA	タ "
YOU, DA	ダ "
YOU, NARU	ナル "
YOU, NAI	ナイ "
SHU	終止形
TAI	連体形で, トキに付く
KA	仮定形で, パに付く
MEI	命令形

接続規則において使用される。

3. 日本語解析・生成部

本章では、入力日本語文に形態素解析処理・構文解析処理を施して日本語 D-tree を得る日本語解析過程、逆に日本語 D-tree に構文生成処理・形態素生成処理を施して日本語文を合成する日本語生成過程の両過程について述べる。ただし本システムではこの構文解析処理と構文生成処理を一つの手続きで行なっているのでこれを構文処理部としてまとめて説明する。

3.1. 日本語形態素処理部⁹⁾

形態素解析処理では、べた書きの漢字平仮名混りの入力日本語文を単語に分割し、そのそれぞれに形態素情報（品詞と単語情報）を付加し単語の宣言の形で出力する。形態素生成処理では、単語の宣言より単語を取り出し、その形態素情報に応じて語形変化を行って一つの文として合成出力する。G_r を構成する接続規則については本処理部において規則適用を行なっている。

3.1.1. 入出力形態

日本語形態素解析処理の入力及び日本語形態素生成処理の出力はべた書きの漢字平仮名混り文である。日本語形態素解析処理の出力及び日本語形態素生成処理の入力は、英語の場合と同様の次の形式である。

(ASSERT (TANGO 開始位置 終了位置 (単語語幹)
(H_{i-1}, C_{i-1}, F_{i-1}, H_i, C_i, F_i)) (単語情報)))

ただし TANGO: PROLOG/KR の述語名
H_i ∈ 品詞, C_i ∈ 活用類, F_i ∈ 活用形

3.1.2. 日本語形態素解析処理

本処理は大きく次の 2 Step から成る。

Step 1: 文字種による分割

Step 2: 最長一致法による単語認定と接続規則適用
以下各 Step について説明する。

[Step 1]: 入力日本語がべた書きの漢字平仮名混り文であるという特徴を生かして、まず初めに、入力日本語文を平仮名から漢字へ変わる部分で区切って幾つかの文字列に分ける。この処理を文字種による分割処理、その分割したそれぞれの文字列を検索単位と呼ぶ。実際この区切り目が単語の変わり目となることが多い⁵⁾ ようである。

[Step 2]: 次に Step 1 で得たそれぞれの検索単位を検索文字列として最長一致法で単語辞書 (Table 10) を検索する。見出し語に活用がある場合には、その活用形を全て生成して検索文字列と比べている。なお、単語辞書の見出し語検索において、次の 2 点を特に考慮している。

(1) 検索を調べる語とその前の語について活用辞書 (Table 8) を参照することにより接続規則を適用し接続を調べている。

(2) 見出し語が漢字平仮名混りの語である場合に

Table 10 A part of word dictionary in Japanese.

((私))	(? ? ?	PRON	() () ()	(1 TAN))
((音楽))	(? ? ?	NOUN	() () ()	(TAN))
((それ))	(? ? ?	PRON	() () ()	(3 TAN))
((好き))	(? ? ?	ADJ	DOU GOKAN)	() ()
((話))	(? ? ?	VERB	SA5 GOKAN)	(JP2 JP3) ()
((よ))	(? ? ?	ADJ	KEI GOKAN)	() ()
((ま))	(VERB ?	YOU. MASU AUX MASU GOKAN)	()	(TEINEI))
((ま))	(AUX ?	IDEN YOU. MASU AUX MASU GOKAN)	()	(TEINEI))
((た))	(VERB ?	YOU. MASU AUX KEI GOKAN)	()	(KIBOU))
((た))	(AUX	IDAN YOU. MASU AUX KEI GOKAN)	()	(KIBOU))

は、次の検索単位まで含めたものを検索文字列として処理を行なっている。これは Step 1 によって単語の途中で不適切に分割された場合を考慮したものである。

なお、本システムで取り扱う日本語には最適な単語分割を行なう為に次のような方針を設けた。

(1) 漢字の使用が不自然でない単語については、なるべく漢字を使用する。

(例 1) 薬が効く、音楽を聞く

(2) 合成語は辞書に登録する。

(例 2) 機械翻訳、蚊取り線香

(3) 慣用句は辞書に登録する。

(例 3) なければならない、はずだ、によって

Table 10 の日本語単語辞書における各項目の構造は次のように表現できる。

((G) (H_{i-1}, C_{i-1}, F_{i-1}, H_i, C_i, F_i) (VP) (T))
ただし

G: 見出し語 (語幹)

活用しない語は、その語そのものを与える。活用する語はその語幹を与える。

H_i ∈ 品詞, C_i ∈ 活用類, F_i ∈ 活用形

VP: 述語機能パターン

述語機能パターンをリスト形式で任意個与えることができる。

T: 単語情報

単語情報をリスト形式で任意個与えることができる。

3.1.3. 日本語形態素生成処理

ここでは入力された単語の宣言中にある単語の接続規則により Table 8 の活用表を参照しながら各単語の語尾を決定し、各単語間の接続を確かめながら一つの文として組み立てて出力する。

3.2. 日本語構文処理部⁶⁾

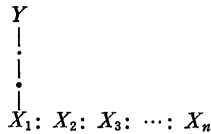
この節では、日本語の単語の宣言より構文解析を行なって日本語 D-tree を出力する日本語構文処理過

程と、逆に日本語 D-tree に構文生成を行なって日本語の単語の宣言を出力する日本語構文生成過程について述べる。

3.2.1. 入出力形態

日本語構文解析過程の入力と、日本語構文生成過程の出力は 3.1.1 項で述べた日本語の単語の宣言である。日本語構文解析過程の出力と日本語構文生成過程の入力は第 2 章で定義した日本語 D-tree である。その表現形式は英語 D-tree とほぼ同様であるが、接続規則は次のように表現している。

接続: Y(X₁ X₂ X₃ ... X_n)



この表現を使った実際の解析例を次に示す。

(例 4) 私はその本を読む。

解析例

JP 3 (SUB (PRON (1) JSUB (2))

DO (DET (3) NOUN (4) JDO (5))

(VERB (6)))

(ASSERT (TANGO 1 2 (私)

(? ? ? PRON () () (1 TAN))

(ASSERT (TANGO 2 3 (は)

(PRON () () JSUB () () ()

(ASSERT (TANGO 3 4 (その)

(? ? ? DET () () (3 TAN))

(ASSERT (TANGO 4 5 (本)

(? ? ? NOUN () () (TAN))

(ASSERT (TANGO 5 6 (を)

(NOUN () () JDO () () ()

(ASSERT (TANGO 6 7 (読)

(? ? ? (JP 3) MA5 SHU) ()

依存の木

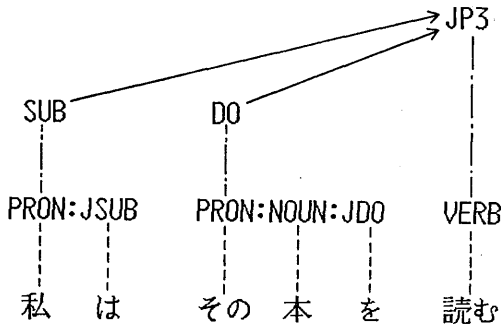


Fig. 1 The dependency tree of example 4.

3.2.2. 日本語構文処理

本処理過程は、英語構文処理過程と比べて、入力文の処理方向が left-to-right から right-to-left になる点と述語機能パターン中の述語 P 以外の機能要素の位置が任意である点を除いては同じである。ここでは日本語依存文法 G_j の初期記号と文法規則を示す。

(1) 初期記号の有限集合を次のように定める。

$$S = \{P\}$$

(2) 機能要素割当規則に Table 11 のようなものを用いている。

(3) 依存規則 R は次の Table 12 と Table 13 の 2 つに分かれる。

4. トランスファ部

本章では、英語 D-tree から日本語 D-tree に変換する英日方向トランスファ処理過程、及び日本語 D-tree から英語 D-tree に変換する日英方向トランスファ処理過程の両過程について述べる。この両過程とも Lisp によって記述している。その為、両過程とも別々のプログラムとなっている。

4.1. 入出力形態

英日トランスファ部の入力英語 D-tree、出力は日本語 D-tree である。逆に、日英トランスファ部の入力は日本語 D-tree、出力は英語 D-tree である。

4.2. トランスファ処理部

トランスファ処理は英、日の D-tree 間の木構造変換および leaf の単語と単語情報の置換を行なう heuristic な処理である。この処理は大きく次の 2 種類に分けられる。

- (1) ある一定の規則に従って変換できる部分
- (2) 規則に当てはまらない特別な処理の部分

Table 11 A part of function assign rule in Japanese.

SUS→PRON	PIV→PRON	IO →NOUN
SUB→NOUN	PIV→NOUN	DO→PRON
P →VERB	PIV→ADJ	DO→NOUN
P →ADJ	IO →PRON	AC→PPRE

Table 12 A part of dependency rule R1 in Japanese.

pattern 1	P (SUB, PIV, *)
pattern 2	P (SUB, *)
pattern 3	P (SUB, DO, *)
pattern 4	P (SUB, IO, DO, *)

Table 13 A part of dependency rule R2 in Japanese.

SUB	(P-NM, *)
P-PM	(P-NM, *)
NO	(P, *)
NC	(P, *)

Table 14 A part of verb pattern conversion dictionary between English and Japanese.

Verb pattern in English	Verb pattern in Japanese
(EP 1-1)	JP 1-1)
(EP 1-1)	JP 1-2)
(EP 1-2)	JP 1-2)
(EP 1-8)	JP 1-2)
(EP 1-4)	JP 1-4)

(1) として次のような規則を設けている。

㊸ 述語 P を構成する動詞の述語機能パターン番号を KEY として Table 14 の述語機能パターン番号対応辞書により対応言語の述語機能パターン番号を得る。このパターン番号と依存規則 R1, R2 によって機能要素の並びを得る。

㊹ 英依存規則 R3 により、日機能要素を構成する日単語の並びと、英機能要素を構成する英単語の依存関係との変換を行う。この時、Table 3 に従い英→日の場合は助詞の追加、日→英の場合は助詞の削除を行う。

㊺ 各単語に対応する単語を Table 15 の単語対応辞書により得る。

(2) としては以下のような処理を上記規則中に必要に応じて埋め込んでいる。

㊻ 日→英のトランスファにおいては必要に応じて関係代名詞 RPRN 「WHICH」を補っている。

Table 15 A part of word conversion dictionary between English and Japanese.

(Word) A part of speech in English	(Word) A part of speech in Japanese
((A) DET	(一つの) DET)
((AN) DET	(一つの) DET)
((ABOUT) DET	(およそ) DET)
((ABOUT) PRE	(についての) PRE)
((AFRICA) NOUN	(アフリカ) NOUN)
((THINK) VERB	(思) VERB)

⑤ 述語機能パターンだけでは対応関係が決まらずに、他の機能要素を構成する単語の品詞も考慮する必要があるもの。

⑥ 機能要素の置換が必要なもの

例えば、英語中の IO 又は DO が日本語中の NO の中の SUB となるもの

⑦ 英日の表現の違いにより木構造が異なるもの

- ・名詞の副詞的用法
- ・態が異なる場合

なお、(2) の処理において、トランスファの前後で単語の数が変わるものについては、単語情報の変換を行っている。

5. 翻訳実験

翻訳実験は本専攻教育用電子計算機 HITAC M-240 H の TSS 下で行った。英語及び日本語の構文処理部のプログラム記述言語は Prolog, 他の処理部のプログラム記述言語は Lisp である。Lisp は UTILISP, Prolog はその UTILISP 上の PROLOG/KR インタープリタを使用した。

実験の入力英語文は中学校英語教科書程度の英文で各英語述語機能パターンを網羅するような代表的な文を選んで使用した。入力日本語文は、入力英語文を入力により訳したものを中心に、各日本語述語機能パターンを網羅する代表的な文を使用した。

以下、まず、依存規則を Prolog でどのように記述するかについて述べ、次に実験に用いた辞書について述べる。最後に翻訳例を示す。

5.1. 依存規則の Prolog による記述

PROLOG/KR ではホーン節を次のように記述する。

unit clause : (ASSERT Q_1)

non-unit clause: (ASSERT $Q_1 Q_2 \dots Q_m$)

question : (AND $Q_2 \dots Q_m$)

ただし ASSERT, AND: PROLOG/KR の予約述語

Q_1, Q_2, \dots, Q_m : goal

又 goal は次のように表す。

(述語 引き数1 引き数2 ... 引き数n)

BIJET においては、各述語の最初の2引数を 3.1.1 項の単語の開始位置、終了位置に対応させており、この2引数を述語間で引渡すことによって入力文の処理範囲を限定し、入力文を left-to-right に処理している。3番目以後の引数の使用方法は特に定めていない。以下、依存規則の PROLOG/KR による記述例を示す。

(例5) 依存規則R1の記述例

P (SUB, *, IO, DO)

↓

(AS (SENTENCE >START >END *PATNO

(*SUB *P *IO *DO))

(SUB >START *N1 *SUB

(*NUMBER *PERSON))

(P *N1 *N2 *P *PATNO

(*NUMBER *PERSON))

(IO *N2 *N3 *IO)

(DO *N3 >END *DO))

この例を見るとわかるように、依存構造を表すリスト表記を引数間で渡すことにより 3.2.1 項と 4.2.1 項で述べた依存構造を組み立てている。

この依存規則の記述の際に問題となる事項に、規則の併合と処理速度の関係がある⁷⁾。本システムでは、似たような規則が数多くある場合に規則の併合を行い処理速度を上げている。ただし、その場合には、規則の併合のために、依存規則と Prolog の記述との対応がとれなくなっている。

5.2. 実験規模

本システムのプログラムサイズを Table 16 に示す。又、形態素処理とトランスファ処理で用いた辞書サイズを Table 17 に示す。

5.3. 翻訳例

英→日の翻訳例を Fig. 2 に、日→英の翻訳例を Fig. 3 に示す。

6. 検 討

BIJET は、Prolog を主体にしてシステムを記述した為、システム開発時間を大幅に節約することがで

Table 16 Program size.

processor		program- ming language	size
English	Morphological analysis	Lisp	260 lines
	Morphological synthesis	Lisp	100 lines
	Syntactic processing	Prolog	154 clauses
Transfer	from English to Japanese	Lisp	127 lines
	from Japanese to English	Lisp	173 lines
Japanese	Morphological analysis	Lisp	330 lines
	Morphological synthesis	Lisp	150 lines
	Syntactic Processing	Prolog	81 clauses

Table 17 Dictionary size

dictionary name	size
English word	216 words
English morphological transformation rule	61 rules
Word conversion	70 words
Verb pattern conversion	22 patterns
Japanese word	94 words
Japanese conjugation	27 types

```

<<<***** START OF TRANSLATION E->J *****>>>
INPUT SENTENCE = (I GAVE HIM THE BOOK)
----- RESULT OF E-PARSING -----
P4
(SUB (PRON (1)) (VERB (2)) IO (PRON (3)) DO (NOUN (DET (4) (5))))
(ASSERT (WORD 1 2 (I) PRON (1 SIN NOM)))
(ASSERT (WORD 2 3 (GAVE) (P1-4 P1-5) (1 SIN PAST)))
(ASSERT (WORD 3 4 (HIM) PRON (3 SIN OBJ)))
(ASSERT (WORD 4 5 (THE) DET ()))
(ASSERT (WORD 5 6 (BOOK) NOUN (SIN)))
----- D-TREE CONVERSION -----
JP4
(SUB (PRON (1)) JSUB (6)) IO (PRON (3)) JIO (7))
DO (PRON (4)) NOUN (5) JDO (8)) (VERB (2) AUX (9)))
(ASSERT (TANGO 6 (が) (PRON (1) JSUB (6) (7) (8) (9))))
(ASSERT (TANGO 7 (に) (PRON (1) JSUB (6) (7) (8) (9))))
(ASSERT (TANGO 8 (を) (NOUN (1) JDO (8) (9) (10))))
(ASSERT (TANGO 5 (本) (??? PRON (1) (2)) (TAN)))
(ASSERT (TANGO 1 (私) (??? PRON (1) (2) (1 TAN NOM)))
(ASSERT (TANGO 3 (彼) (??? PRON (1) (2) (3 TAN OBJ)))
(ASSERT (TANGO 4 (その) (??? DET (1) (2) (3))))
(ASSERT (TANGO 2 (与え) (??? (JP4 JPS) IDAN YOU.TA) (1)))
(ASSERT (TANGO 9 ( ) (VERB ? YOU.TA AUX TA SHU) (KAKO)))
----- RESULT OF GENERATING -----
OUTPUT SENTENCE = (私が彼にその本を与えた)
TIME CONSUMED FOR THIS TRANSLATION = 714 MSEC
<<<***** START OF TRANSLATION E->J *****>>>
INPUT SENTENCE = (I GIVE THE BOOK TO HER)
OUTPUT SENTENCE = (私がその本を彼女へ与える)
TIME CONSUMED FOR THIS TRANSLATION = 1427 MSEC
<<<***** START OF TRANSLATION E->J *****>>>
INPUT SENTENCE = (I THINK IT BEAUTIFUL)
OUTPUT SENTENCE = (私がそれを美しいと思う)
TIME CONSUMED FOR THIS TRANSLATION = 541 MSEC
<<<***** START OF TRANSLATION E->J *****>>>
INPUT SENTENCE = (I TOLD HIM THAT HER NAME IS JANE)
OUTPUT SENTENCE = (私が彼に彼女の名前がジェーンですと
言った)
TIME CONSUMED FOR THIS TRANSLATION = 1217 MSEC
    
```

Fig. 2 Translation examples from English to Japanese.

```

<<<***** START OF TRANSLATION J->E *****>>>
INPUT SENTENCE = (私は彼にその本を与えた)
***** RESULT OF J-PARSING *****
JP4
(SUB (PRON (1)) JSUB (2)) IO (PRON (3)) JIO (4))
DO (DET (5)) NOUN (6) JDO (7)) (VERB (8) AUX (9)))
(ASSERT (TANGO 1 2 (私) (??? PRON (1) (2) (1 TAN)))
(ASSERT (TANGO 2 3 (は) (PRON (1) JSUB (2) (3) (4) (5))))
(ASSERT (TANGO 3 4 (彼) (??? PRON (1) (2) (3 TAN)))
(ASSERT (TANGO 4 5 (に) (PRON (1) JSUB (2) (3) (4) (5))))
(ASSERT (TANGO 5 6 (その) (??? DET (1) (2) (3))))
(ASSERT (TANGO 6 7 (本) (??? NOUN (1) (2) (TAN)))
(ASSERT (TANGO 7 8 (を) (NOUN (1) JSUB (2) (3) (4) (5))))
(ASSERT (TANGO 8 9 (与え) (??? (JP4 JPS) IDAN YOU.TA) (1)))
(ASSERT (TANGO 9 10 ( ) (VERB ? YOU.TA AUX TA SHU) (KAKO)))
***** D-TREE CONVERSION *****
P4
(SUB (PRON (1)) (VERB (8)) IO (PRON (3)) DO (NOUN (DET (5) (6))))
(ASSERT (WORD 6 (BOOK) NOUN (SIN)))
(ASSERT (WORD 1 (I) PRON (1 SIN)))
(ASSERT (WORD 3 (HIM) PRON (3 SIN)))
(ASSERT (WORD 5 (THE) DET ()))
(ASSERT (WORD 8 (GAVE) (P1-4 P1-5) (PAST)))
***** RESULT OF GENERATING *****
OUTPUT SENTENCE = ( I GAVE HIM THE BOOK )
TIME CONSUMED FOR THIS TRANSLATION = 2505 MSEC
<<<***** START OF TRANSLATION J->E *****>>>
INPUT SENTENCE = (私はその本を彼女へ与える)
OUTPUT SENTENCE = ( I GIVE THE BOOK TO HER )
TIME CONSUMED FOR THIS TRANSLATION = 4274 MSEC
<<<***** START OF TRANSLATION J->E *****>>>
INPUT SENTENCE = (私はそれを美しいと思う)
OUTPUT SENTENCE = ( I THINK IT BEAUTIFUL )
TIME CONSUMED FOR THIS TRANSLATION = 2102 MSEC
<<<***** START OF TRANSLATION J->E *****>>>
INPUT SENTENCE = (私は彼に彼女の名前はジェーンですと言った)
OUTPUT SENTENCE = ( I TOLD HIM THAT HER NAME IS JANE )
TIME CONSUMED FOR THIS TRANSLATION = 4907 MSEC
    
```

Fig. 3 Translation examples from Japanese to English.

きた。又、構文処理部を解析と合成の両方に用いることができたので、モジュールの削減を計ることができた。本システムの開発を通じて、Prolog の種々の特徴を確認したが、処理速度については、大学研究所レベルではほぼ満足のいくものであった。

BIJET における D-tree の基本的な考え方は KULTS (Kyushu University Language Translation System) と同じであるが、ここでは D-tree を pivot language としてとらえるのではなく、トランスファ方式における中間表現としてとらえることにより表層レベルの翻訳システムとしてすっきりしたものとなった。本システムにおいて使用した言語情報は、英語に関しては KULTS のものを基本とし、日本語に関しては今回新たに抽出したものである。日本語言語情報に関しては今後実験を重ねながら更に検討を加える必要がある。又、日本語処理部に加えた制限についても、今後システムの改良を行って完全に消失するようにするのが望ましい。

7. む す び

BIJET は表層の構文情報のみを使用するトランスファ方式の英日双方向機械翻訳システムとして試作した。翻訳実験の結果、構文情報のみではトランスファできない文も存在することを確認した。これらの文を翻訳する為には意味情報の利用が不可欠であるが、意味マーカのような簡単なものから、横田らの意味表現のような高度なものまで色々ある意味処理をいつ、どのような形で取り入れるのが最良なのかは、今後検討を加える必要がある。又、今回英→日方向と日→英方向の両方向の翻訳を行ってみて、日本語は構文上色々な表記が許されているため、意味処理が入っていない構文処理の結果だけから英語へトランスファを行うのはかなり無理があることがわかった。しかし、Prolog という便利な道具を得て理論の実践が容易になったので、今後理論的検討と実験を繰り返していきたい。

なお、本研究の一部は昭和57~59年度文部省科学研究費 特定研究「言語の標準化」(57215006, 58207009, 59101006) の援助を受けたものである。

参 考 文 献

- 1) 吉武, 三池, 田町: “日英双方向機械翻訳システム BIJET の概要と英語処理について”, 九大総理工研究科報告, Vol. 6, No. 2 (昭和60年).
- 2) 吉武春光: “D-tree モデルに基づく英日機械翻訳システム”, 昭和56年度, 九大総理工情報システム学専攻修士論文.
- 3) Toru Hitaka and Sho Yoshida: “A SYNTAX PARSER BASED ON THE CASE DEPENDENCY GRAMMAR AND ITS EFFICIENCY”, Proc. of COLING 80, pp: 295-302 (1980).
- 4) 末廣陽一: “英日双方向翻訳システムの形態素処理部について”, 昭和58年度, 九大情報工学科卒業論文.
- 5) 「特集: 日本語情報処理」, 情報処理, Vol. 20, No. 10 (1979).
- 6) 三池誠司: “PROLOG による英日双方向翻訳の試み”, 昭和58年度, 九大総理工情報システム学専攻修士論文.
- 7) 安部: “文の解析例から Prolog プログラムを生成する方法について”, bit, Vol. 15, No. 12 (1983).