TR-IT-0088

TDMT 翻訳知識作成ツールの利用方法

河井 淳

Jun KAWAI

1994 年 12 月

概要

TDMT （変換主導機械翻訳）の翻訳知識の作成および簡易の翻訳実験を行なうためのツールについて報告する。本ツールは emacs 19.27 の mule-2.1 上に作られ、ILISP 5.6、Lucid Common Lisp 4.1 を利用しており、TDMT-MULTI-4 のシステムを対話的に動作させながら効率的に翻訳知識を作成することができる。主な機能としては、翻訳知識の検索、翻訳実験、適用された翻訳知識および文構造の表示などがある。

エイ・ティ・アール音声翻訳通信研究所
ATR Interpreting Telecommunications Research Laboratories

目次

第 1 章

はじめに

1.1 TDMT-MULTI システムの概要

TDMT-MULTI システムは変換主導型機械翻訳（TDMT）[古瀬, 94][Furuse, 94]の多言語翻訳実験システムであり、単方向の翻訳処理を切替えることによって多言語間の翻訳処理を実現することができます。TDMT-MULTI システムは形態素解析、変換、生成の 3 つの主要なモジュールから構成されています。翻訳処理を実現するには原言語の形態素解析と目的言語の生成モジュールを用意し、さらに変換知識などの翻訳知識を各翻訳方向（原言語から目的言語）で用意する必要があります。

1.2 TDMT 翻訳知識作成ツールの概要

TDMT 翻訳知識作成ツールは TDMT-MULTI システム用の翻訳知識を効率よく作成するためのツールです。本ツールの提供する主な機能としては、翻訳知識の検索、翻訳実験、適用された翻訳知識および文構造の表示などがあります。なお、本ツールは emacs 19.27 の mule-2.1 上に作られ、ILISP 5.6、Lucid Common Lisp 4.1 を利用しています。

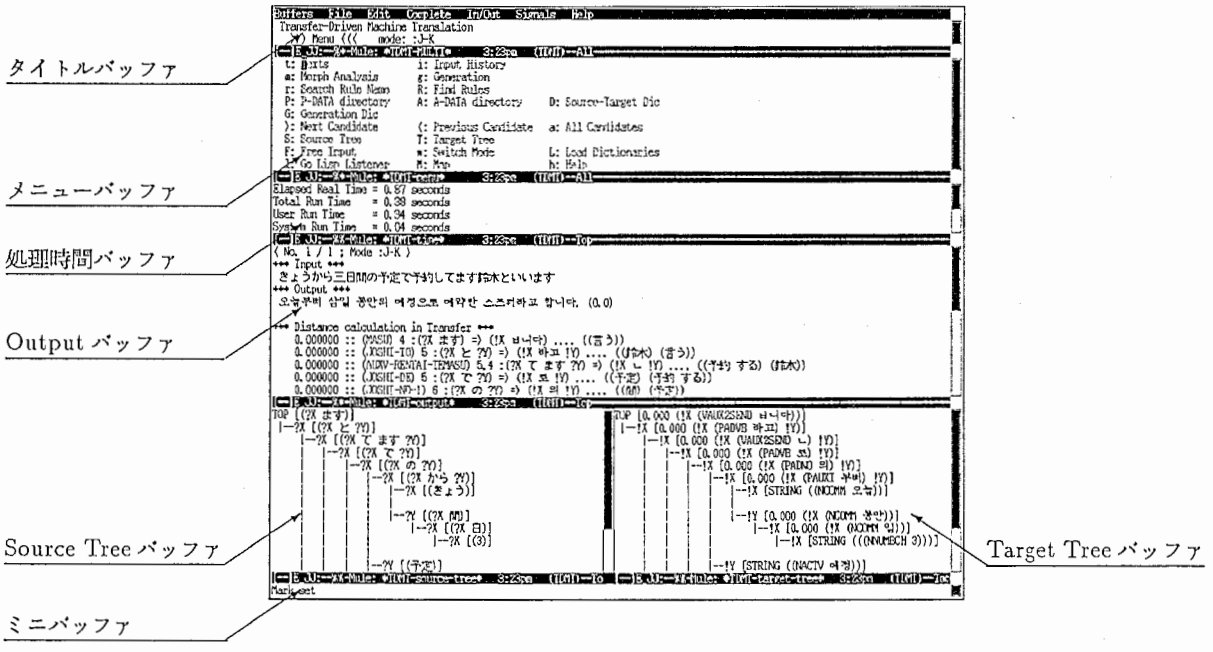

図 1.1: TDMT 翻訳知識作成ツール

1.3 TDMT の翻訳知識について

本ツールは主として TDMT の翻訳知識を作成するためのものです。TDMT の翻訳知識とは、TDMT システムの変換処理（transfer モジュール）が使用するルールおよび辞書のことで、これらは翻訳処理の中心的な部分で使われます。これらの知識は相互に依存する関係をもっていますので、そのバランスを考慮しながら作成する必要があります。ここでは、TDMT の翻訳知識の種類と働きについて簡単に述べます。

- p-data
  変換ルールのことを指します。変換処理はこの変換ルールを組み合わせて入力文の構造をとらえ、翻訳の骨組みを決めます。各ルールは一つの原言語パターンと一つ以上の目的言語パターンからなります。目的言語パターンはそれぞれいくつかの用例を持っており、この用例と実際の入力との比較により適切な目的言語パターンを選び翻訳するしくみです。

```
(define-pattern degozaimasuka :j-e 4
  (?x "で" "ございます" "か")(:head 1)
  =>
  (((!x) (yn-q 1))
   (("予定"))
    (("何"))
    )
   )
  (("that" "is" "right") (yn-q 1 2 3))
   (("さよう"))
    )
   )
 )
```

- a-data
  解析ルールのことを指します。変換処理の前処理として形態素列の調整やマーカーの挿入などを行ないます。これは次の 3 種類に分類されます。

  1. Lexical transformation
     形態素を調整するためのルールです。主に形態素の合成などに使います。

```
(define-morph @add-suru :j-e (1 する)
  (:all-copy . 1)(:pos . 本動詞))
(define-lexical-transformation sahen-verb-wo :j-e 3
  (
   (((:pos . サ変名詞)) ((:word . "を"))((:reg-exp . "する")) )
   =>
   ((@add-suru)
    )
   )
  )
```

2. Local transformation
   形態素列の間にマーカーを挿入するためのルールです。

```
(define-local-transformation auxv/rentai-cn :j-e 3
  ((((:pos . 助動詞) (:conj-form . 連体)) ((:pos . 普通名詞)) ((:pos . 助動詞)))
   =>
   ((1 <auxv/rentai-cn-auxv> 2 3)
    (("たい" "わけ" "です")
     )
    )
  )
  ((((:pos . 助動詞)(:conj-form . 連体)) ((:pos . 普通名詞)))
   =>
   ((1 <auxv/rentai-cn> 2)
    (("たい" "こと")
     ("た" "パンフレット")
     ("た" "段階")
     )
    )
  ))
```

3. Total transformation
   形態素列を並べ換えるためのルールです。

- source to target dictionary
  変換辞書のことを指します。変換ルールを繰り返し用いた結果できる木構造の末端部分の単語を翻訳するための辞書です。

```
(define-dic japanese-to-english-dic t
;; there
 ("ございません" (there "there")(adv "not"))
;; pronoun 代名詞
 ("私" (pron "I"))
 ("お客様" (pron "you"))
 ("こちら" (pron "this"))
 ("これ" (pron "this"))
 ("そちら" (pron "you"))
 ("それ" (pron"it"))
 ("そこ" (pron "there"))
 ("当社" (pron "we"))
 ("やつ" (pron "one"))
 ...
 )
```

第 2 章

システムのセットアップと終了

ここでは、システムを使用するにあたって必要な環境設定はすべて整っている状態でのシステムのセットアップについて説明します。環境設定については第5章を参照してください。

2.1 システムのセットアップ

1. unix のシェルまたはシステムメニューから mule-2.1 を立ち上げる。（下線部をタイプして下さい。）

```
% mule-2.1 &
```

2. mule 上で TDMT-MULTI を立ち上げる。（次のコマンドをタイプすると Lisp が立ち上がります。このときカーソルはミニバッファにあります。）

```
M-x run-tdmt
```

3. ミニバッファにセットアップファイルを指定する。（Lisp が立ち上がったのを確認してから [RET] 入力して下さい。）これにより TDMT-MULTI システムの各モジュールおよび翻訳知識などのデータがロードされます。ロードされているファイルのディレクトリが正しいかどうか確認して下さい。なお、セットアップファイルのパス名は各ユーザーごとに異なりますので注意してください。

```
~ /setup.lisp                ... セットアップファイルのパス名
```

4. ロードが終了すると、Lisp のプロンプトがでた状態になります。その後、次のコマンドをタイプして、TDMT 翻訳知識作成用の画面を呼び出して下さい。

```
M-0                  ... (Meta + ゼロ)
```

以上でセットアップは完了です。また、先に Lisp（M-x lucid）と TDMT（> (load "~ /setup.lisp")）を立ち上げて、後で TDMT 翻訳知識作成ツールをロードすることもできます。その時は、M-x tdmt-windows コマンドを入力してください。

2.2 システムの終了

基本的な終了方法を説明します。

1. TDMT および Lisp の終了： M-x lucid または後述の l コマンドにより Lisp listener バッファを呼び出し、次のようにタイプして [RET] する。

```
> (quit)
```

2. mule の終了：編集したファイルをすべてセーブするかバッファを kill してから次の key をタイプする。

```
C-x C-c
```

Lisp をセーブせずに mule を終了しようとした場合には、

```
Active processes exist; kill them and exit anyway? (yes or no)
```

というメッセージがミニバッファに表示されます。この場合は yes[RET] と入力すれば終了します。また、変更されていてセーブされていないファイルがある場合には、

```
Save file ファイル名? (y, n, !, ., q, C-r, C-h)
```

というメッセージがミニバッファに表示されます。この場合は y と入力すればセーブされます。

多くのファイルバッファをセーブしたり kill する必要のあるときは次の方法が便利です。

- ファイルをまとめてセーブする方法
  1. C-x C-b で *Buffer-List* を呼び出す。
  2. セーブするファイル名の行で s と入力して S マークをつける。
  3. x と入力すると、S マークついたすべてのファイルがセーブされる。
- バッファをまとめて kill する方法
  1. C-x C-b で *Buffer-List* を呼び出す。
  2. セーブするファイル名の行で d と入力して D マークをつける。
  3. x と入力すると、D マークついたすべてのファイルが kill される。

第 3 章

コマンド操作説明

3.1 基本操作説明

TDMT 翻訳知識作成用の画面（以下 TDMT 画面という。）での基本的な操作について説明します。この画面は “*TDMT-” という接頭辞付きの名前を持ったいくつかのバッファ（以下、TDMT バッファという。）で構成されており、それらのバッファ上では通常 mule で使用するものとは違う key が割り当てられています。TDMT バッファは各バッファごとにそれぞれ固有の特殊な key が割り当てられていますが、ここでは TDMT バッファに共通する key について説明します。

1. カーソルの移動やカット、ペーストなどの基本操作は本来の mule とほとんど同じです。

2. TDMT 専用に割り当てられた key は Ctr, Meta key を必要としないものが多いですが、Shift key はよく使います。以下の key の説明で大文字のものは Shift key 入力が必要です。

3. マウスのミドルクリックはその場所にカーソルを合わせ [RET] をタイプしたのと同じです。基本的には紫に色付けされた部分がマウスセンシティブでマウスがセンスすると緑にハイライトされますので、そこでミドルクリックすると割り当てられたコマンドが起動されます。

4. TDMT バッファに共通する key コマンドのうち特殊なものを以下に示します。特に初めの 2 つは頻繁に使うことと思われますので、その動作をよく理解して下さい。

[key]	[機能]
M-0	Meta+ ゼロ。TDMT 画面を抜ける。再度タイプすると TDMT 画面を呼び出す。
M-q	現在の TDMT バッファから抜ける。
o	別のウィンドウにカーソルを移動する。
d	現在カーソルのあるウィンドウを広げる。またはもとに戻す。
b	TDMT バッファを選択する。
n	カーソルを下方移動する。
p	カーソルを上方移動する。

5. 次節で説明するメニューバッファに表示される各コマンドも基本的に共通です。

3.2 メニュー

TDMT-menu （メニューバッファ）

[key]	[機能]
t	入力テキストの呼び出し。
i	入力履歴の呼び出し。
b	TDMT バッファの呼び出し。
m	形態素解析処理結果の表示。
g	生成処理結果の表示。
r	Output バッファ上でのルール名の検索。
R	ロードされているルール定義の呼び出し。
P	ロードされた p-data のディレクトリ表示。
A	ロードされた a-data のディレクトリ表示。
D	ロードされた変換辞書の呼び出し。
G	ロードされた生成辞書の呼び出し。
>	翻訳結果の次候補表示。
<	翻訳結果の前候補表示。
a	翻訳結果の全候補表示。
S	Source tree の表示。
T	Target tree の表示。
F	自由入力。
w	翻訳モードの切替え。
L	ルール、辞書類のロード。
l	Lisp listener の呼び出し。
M	システムマップの表示。
h	ヘルプ表示。

TDMT 用のコマンドが表示されているバッファをメニューバッファといいます。このバッファは TDMT 画面を呼び出した直後に必ず表示されるバッファで、上記の key 入力の他に、カーソルを移動して [RET] 入力するか、マウスのミドルクリックによってもコマンドを選択することができます。（なお、このバッファ上ではカーソルの移動は [SPC]、[DEL] によっても行なえます。）

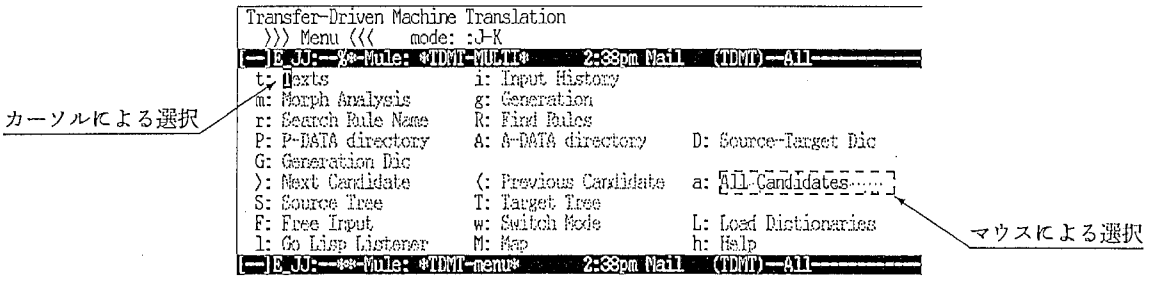

図 3.1: メニューバッファ

3.3 翻訳結果の表示について

翻訳結果は次の各バッファに表示されます。結果に複数候補が存在する場合は、第一位の候補に関するものが表示されます。他の候補の情報をみる場合には、>、<、aコマンド（3.9節参照）により他候補を選択して下さい。

1. 処理時間バッファ：入力から出力までの処理時間を表示する。

Elapsed Real Time	実際に経過した時間
Total Run Time	User + System Run Time
User Run Time	翻訳処理時間
System Run Time	Lisp システムが使用した時間

2. Output バッファ：処理の詳細を表示する。バッファの先頭に候補番号が表示され、次に入力と出力が表示される。また、適用されたルールの情報と処理過程が表示される。

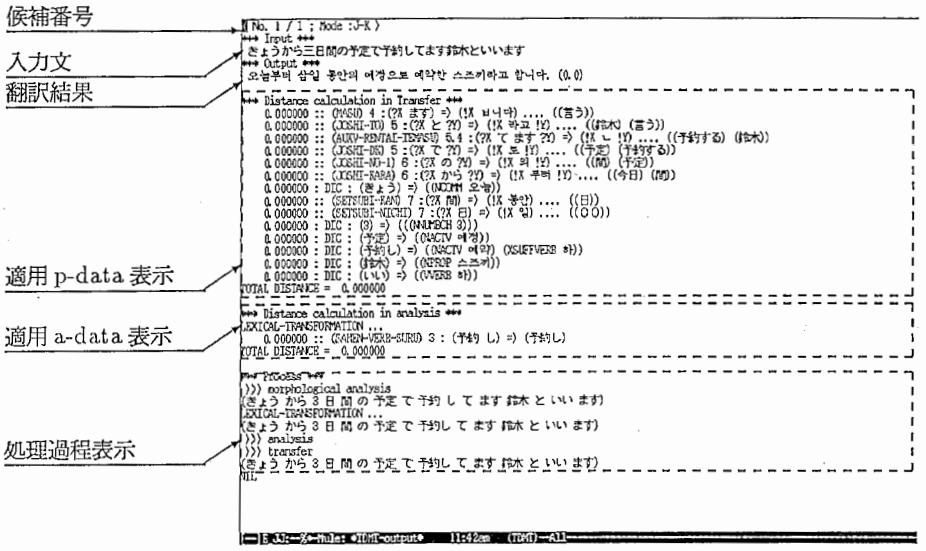

図 3.2: Output バッファ

3. Source tree バッファ、Target tree バッファ：ルールの適用状況を木構造で表示する。

3.4 入力用テキストのロード（t コマンド）

TDMT-menu （メニューバッファ）
[key] [機能]
t 入力テキストの呼び出し。

テキストの呼び出しをします。システム立ち上げ後、初めてこのコマンドが使われた時は、ファイルからテキストをロードしますので少し時間がかかります。

1. テキストの大分類が表示されますので、カーソルを移動して [RET] で選択して下さい。
   TDMT-text-1 （テキスト大分類バッファ）
   [key] [機能]
   RET 大分類の選択。

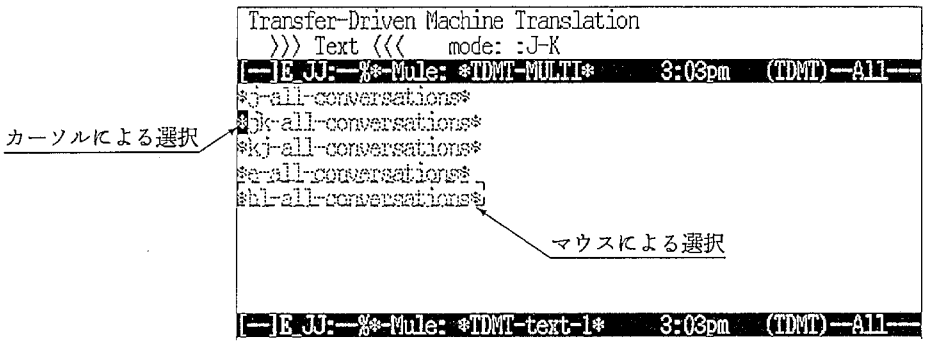

図 3.3: テキスト大分類バッファ

2. テキストの小分類が表示されますので、カーソルを移動して [RET] で選択して下さい。
   TDMT-text-2 （テキスト小分類バッファ）
   [key] [機能]
   RET 小分類の選択。
   t 翻訳のバッチテスト。
   ここでの t コマンドはテキストをまとめてバッチテストするコマンドで、結果の出力先はバッファ、ファイル、Lisp listener のいずれかの中から選択して下さい。

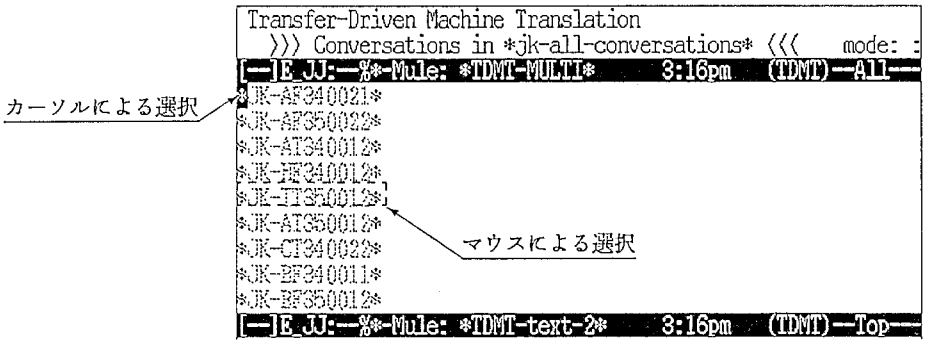

図 3.4: テキスト小分類バッファ

3. テキストが表示されますので、カーソルを移動して[RET]入力するとカーソル行の翻訳が実行されます。

TDMT-text-3 （テキストバッファ）

[key]	[機能]
RET	一文翻訳実行。
C-f	カーソル前方移動、入力変更モード＊への切替え。
C-b	カーソル後方移動、入力変更モード＊への切替え。

図 3.5: テキストバッファ

＊ここで入力変更モードを使用した場合、実際にはテキストが書き換えられることはありません。ここでの入力変更はあくまでも一時的なもので、再度メニューバッファから呼び出したときには元のテキストが表示されます。

3.5 入力履歴の参照（i コマンド）

TDMT-menu （メニューバッファ）

[key]	[機能]
i	入力履歴の呼び出し。

一度翻訳実行された入力文が入力履歴バッファに表示されます。最後に翻訳された入力が先頭に表示されます。カーソルを移動して [RET] 入力、またはマウスのミドルクリックによって翻訳が実行されます。

TDMT-input-history （入力履歴バッファ）

[key]	[機能]
RET	一文翻訳実行。
D	履歴からカーソル行を削除。
C	履歴をすべて削除。
C-f	カーソル前方移動、入力変更モードへの切替え。
C-b	カーソル後方移動、入力変更モードへの切替え。

また、この入力履歴バッファは基本的に文の書き換えができないモードですが、C-f, C-b のコマンドによってカーソルを移動した後には、文を書き換えることができるようになります。この機能は以前に入力した文を少し変更して再度入力したい場合に便利です。

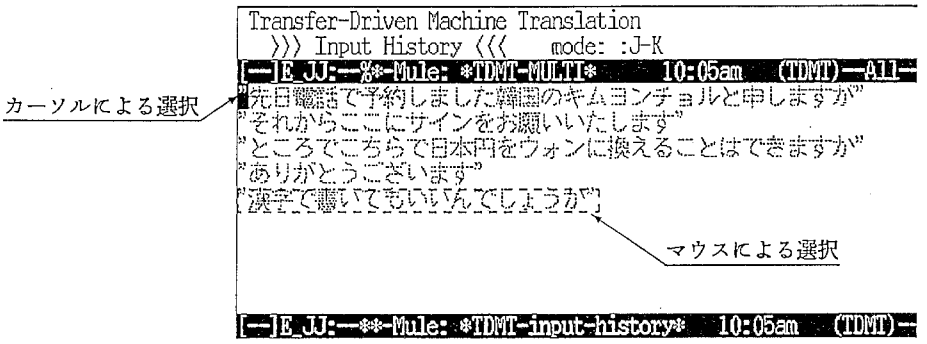

図 3.6: 入力履歴バッファ

3.6 形態素解析処理情報の参照と部分翻訳（m コマンド）

TDMT-menu （メニューバッファ）
[key] [機能]
m 形態素解析処理結果の表示。

最後に翻訳実行された文の形態素解析結果を表示します。同じ文に対して続けて m コマンドが使用された場合は、形態素解析処理の再計算をせずに形態素解析処理バッファを呼び出すだけです。

TDMT-morph-analysis （形態素解析処理バッファ）
[key] [機能]
RET 形態素解析処理結果の部分翻訳実行。

このバッファには形態素解析結果のうちの一部分を翻訳する機能があります。範囲の指定は通常の emacs での region 指定と同様に行ないます。

1. まず、指定する部分の先頭形態素の行頭にカーソルを移動する。
2. ここで C-[SPC] と入力するとミニバッファに"Mark set" と表示される。
3. 指定する部分の最後の形態素の次の行までカーソルを移動する。
4. ここで [RET] 入力すると指定部分が翻訳されます。

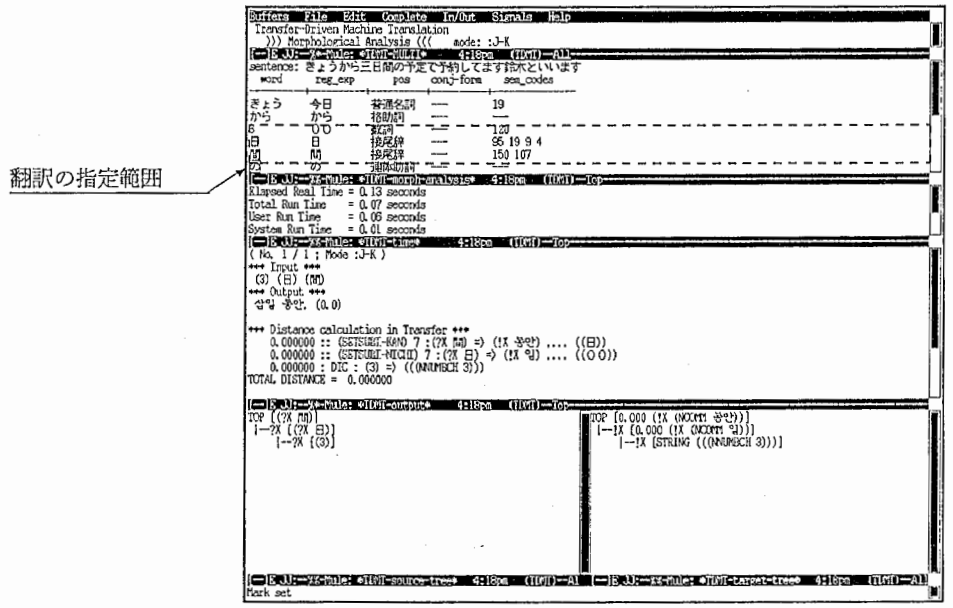

図 3.7: 形態素解析処理バッファ

3.7 生成処理情報の参照（g コマンド）

TDMT-menu （メニューバッファ）
[key] [機能]
g 生成処理結果の表示。

最後に翻訳実行された文の生成処理情報を表示します。生成処理モジュールへの入力と出力が表示されます。変換ルール（p-data）で与えた生成情報が意図したとおりになっているかどうかを確認できます。

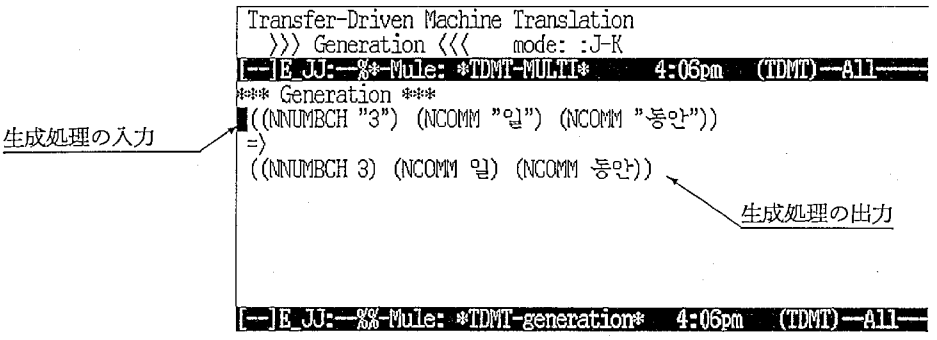

図 3.8: 生成処理バッファ

3.8 翻訳知識の検索、編集、再ロード、定義の無効化（r、R、Lコマンドなど）

TDMT-menu（メニューバッファ）

[key]	[機能]
r	Outputバッファ上でのルール名の検索。
R	ロードされているルール定義の呼び出し。
P	ロードされたp-dataのディレクトリ表示。
A	ロードされたa-dataのディレクトリ表示。
D	ロードされた変換辞書の呼び出し。
G	ロードされた生成辞書の呼び出し。
L	ルール、辞書類のロード。

翻訳知識（ルールおよび辞書）を検索したり、編集するためのコマンドです。DまたはGコマンドは直接該当ファイルのバッファを呼び出します。ルールについては多くのファイルをアクセスする必要があるので、操作が少し複雑になります。一般的なルールの編集手順は次のようになります。

3.8.1 一般的なルールの編集手順

1. 目的の文を翻訳実行してみる。
2. Outputバッファにルール名が表示されている場合はrコマンドで問題のあるルール名にカーソルを移動して、[RET]入力するとファイルのバッファが呼び出される。（マウスのミドルクリックでもよい。）
3. Outputバッファに結果が出ない場合は、Pコマンドまたは Aコマンドによりディレクトリを見て、ファイル名からルールを探す。
4. キーワードからルールを探したい場合や、新しいルールを追加したい場合は後述のルール検索コマンド（Rコマンド）を使用して下さい。
5. ルールを修正して、C-c C-cにより再ロードする。このとき、ルール名を確認し、エラーメッセージがでていないかどうかも確認して下さい。ルールに問題がある場合は修正した後に再度ロードして下さい。デバッガに落ちてしまった場合はその復旧も行なって下さい。
6. TDMT画面に戻り、再度翻訳実行してみて、結果がよくなるまで上記の操作を繰り返す。
7. ある程度ルールが決まったところでファイルをセーブする。

3.8.2 ルール検索と新規ルールファイル作成のしかた（R コマンド）

キーワードからルールを検索したい場合や、新規にルールファイルを作成したい場合には R コマンドを使用します。

1. R コマンドを起動するとミニバッファにルールの種別を確認してくるので、p-data, lexical, local, total の中から選択する。[RET] 入力するとディフォルトの種別となる。
2. p-data の場合はキーワードを入力する。
3. lexical, local, total の場合はプライオリティまたは all を入力する。
4. Output バッファに検索されたルールの一覧が表示されるので、ルール名にカーソルを移動して [RET] 入力するとファイルのバッファが呼び出される。
5. また、新規にファイルを作成したい場合は、Output バッファの中の *NEW-P-DATA-RULE* を選択してファイル名を入力して新しいファイルを作成する。

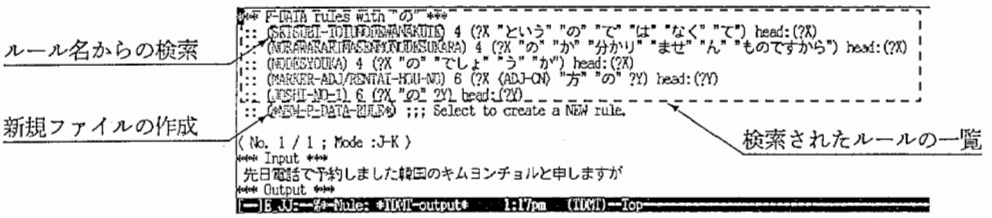

図 3.9: ルール検索

3.8.3 再ロードのしかたとルール定義の無効化（Lコマンド）

ファイルにセーブされたルールをすべて再ロードするにはLコマンドを使用します。このコマンドはセーブされているファイルのみをロードしますので、セーブし忘れたファイルがないかあらかじめ確認して下さい。また、ルールをファイルから抹消した場合には必ずこのコマンドを使って古い定義を無効にして下さい。

1. Lコマンドを起動すると *TDMT-dictionaries* バッファが呼び出される。
2. カーソルを移動して [RET] 入力すると該当するルールや辞書がロードされる。このとき Lisp listener にロードされるファイル名などが表示されるので確認して下さい。
3. ロードはモジュールごとに行なわれ、以前にロードした内容は一度消去されます。
4. また、ファイルに問題があって正しくロードできないか、または途中でデバッガにおちる場合がありますのでその時はファイルを修正したのちに再度ロードして下さい。（デバッガからの復旧も忘れず行なって下さい。4.3節参照。）

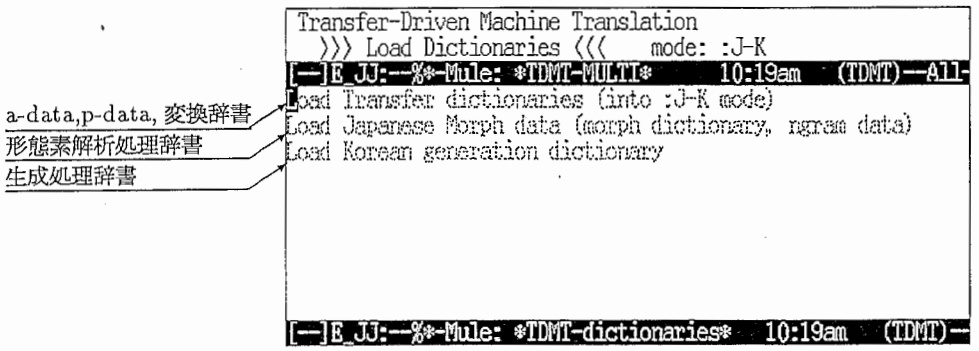

図 3.10: TDMT-dictionaries バッファ

3.9 他候補選択（>、<、a コマンド）

TDMT-menu （メニューバッファ）

[key]	[機能]
>	翻訳結果の次候補表示。
<	翻訳結果の前候補表示。
a	翻訳結果の全候補表示。

翻訳結果が複数個存在する場合には Output バッファや Source tree バッファ、Target tree バッファなどに表示される情報は第一位の候補のものです。そこで上記のコマンドを使用して他の候補の情報を見ることができます。a コマンドではカーソルを移動して [RET] することにより、候補を選択することができます。

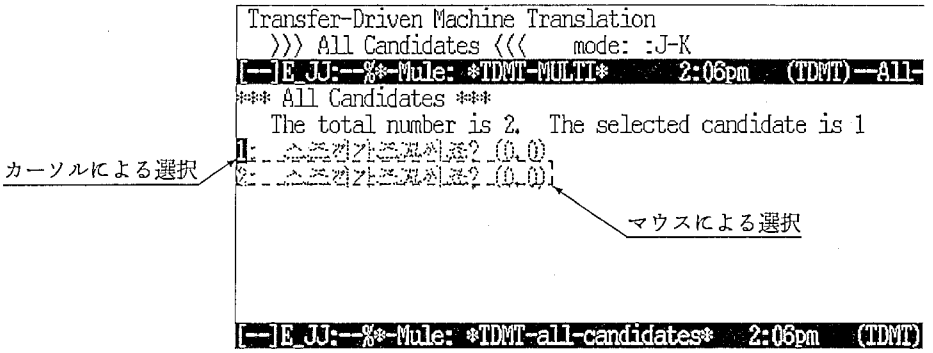

図 3.11: 候補選択バッファ

3.10 ユーティリティコマンド

TDMT-menu （メニューバッファ）

[key]	[機能]
b	TDMT バッファの呼び出し。
S	Source tree の表示。
T	Target tree の表示。
F	自由入力。
w	翻訳モードの切替え。
l	Lisp listener の呼び出し。
M	システムマップの表示。
h	ヘルプ表示。

1. b コマンドは任意の TDMT バッファを呼び出します。ミニバッファで呼び出すバッファを指定して下さい。
2. S コマンド、T コマンドはそれぞれの tree が表示されているバッファを広げます。
3. F コマンドはミニバッファからの自由入力を翻訳します。
4. w コマンドは翻訳方向のモードを切替えます。
5. l コマンドは Lisp listener を呼び出します。M-x lucid コマンドと同じです。
6. M コマンドは現在TDMT システムがどのようなファイルやディレクトリを使用しているかを確認したり、一時的にパスを切替えたりできます。
7. h コマンドは現在カーソルのあるバッファについてTDMT に関わるコマンドの key binding を表示します。[SPC] 入力により順次その説明をみることができます。

第 4 章

使用上の注意

4.1 文の入力と翻訳実行について

翻訳のための文を入力する方法は次の 3 通りがあります。

1. t コマンドによりテキストを呼び出して選択または修正入力する。
2. i コマンドにより入力履歴を呼び出して選択または修正入力する。
3. F コマンドによりミニバッファから入力する。

修正入力とは、もとの文を書き換えて入力することですが、これを使うと形態素解析の結果が書き換え部分以外でも変わってしまう場合があります。そこで、単に部分的な翻訳結果をみたいという場合には修正入力は使わずに、m コマンドによる部分翻訳の利用をおすすめします。(3.6節参照)

また、本ツール上で翻訳を実行した場合、経過実時間 (Elapsed Real Time) が大きくなり、翻訳処理に必要な時間 (User Run Time) の 2 倍程度かかってしまいます。これは本ツールと Lisp のインターフェースの問題が原因と思われ、現在解決方法を模索中です。高速な処理が必要な場合は、Lisp listener 上で次の関数を実行して下さい。実行が終了した後で TDMT 画面にもどり、a コマンド (3.9節参照) で候補選択をすれば必要な情報がみれます。

```
> (time (translate "ありがとうございます"))
```

4.2 ファイル編集について

辞書やルールはほとんどが括弧をもとにデータを認識するようになっているので、括弧の対応には十分注意してください。間違えた場合は、ロード時にエラーがでたり、正常にロードできなくなります。また、間違え方によってはロード時にエラーが検出できない場合もあります。

- 括弧の対応を調べる方法 1
  1. チェックする" ("括弧の位置にカーソルをあわせる。
  2. M-C-f とすると括弧をスキップして、対応する") " 括弧の後ろにカーソルが移動する。
  3. M-C-b とすると逆向きに括弧をスキップする。
- 括弧の対応を調べる方法 2
  1. 範囲 (region) 指定する。
  2. M-x find-unbalanced-region-lisp を使って指定範囲の括弧の対応を調べる。
  3. または M-x find-unbalanced-lisp を使ってバッファ全体の括弧の対応を調べる。

4.3 エラーについて

本ツールを使用しているうちに様々なエラーメッセージに遭遇する場合があります。これにはいろいろな原因があり、それを特定するのが困難な場合もあります。また、これを無視して操作を続けても問題なく動作するようにみえる場合もあります。しかし、一般にはこれを放置しておくと後で問題がさらに深刻化することになりますので、できる限りその場その場で解決しながら先へ進む方が好ましいと思われます。ここでは、エラーメッセージが発生した場合の原因の見つけかたと、エラーの復旧のしかたについて述べます。

- エラーの原因

  1. ルールや辞書のロード時のエラー。
     これは括弧の対応があっていない場合や、必要な要素が抜けている（あるいは多い）場合など書式上のエラーが原因の場合がほとんどです。また、何らかの理由で Lisp のパッケージが transfer 以外のものになってしまって、ロードできない場合もあります。この疑いのあるときは C-c p コマンドで現在のパッケージを確認して、もし違っていれば Lisp listener 上で (in-package 'transfer) を実行してから再度ロードを試みてください。
  2. 翻訳実行時のエラー。
     これはプログラムのバグの可能性もありますが、ルールや辞書にエラーがあって、ロード時にエラー検出ができなかった場合にも発生することがあります。
  3. Lisp Listener 上に"Return from break" のメッセージが出る。
     これはシステムに対して同時に二つ以上の仕事を要求した場合（例えば、ルールのロード中に翻訳実行した場合など）に出ます。このメッセージがでている時は、システムの動作が意図したとおりにならない場合（例えば、ロードしたはずのルールが翻訳結果に反映されていないなど）がありますので注意が必要です。
  4. その他の警告メッセージが出る。
     ルールのロード時に出る警告メッセージは、ルールのエラーを指摘している場合がありますので、十分注意してください。
  5. Lisp listener に入力しても受け付けない。
     Lisp 自体がおかしくなっている場合があります。原因は翻訳処理などが重すぎて負荷がかかりすぎていることなどが考えられます。

- エラーからの復旧

  1. Lisp listener のプロンプトが"－ >" になっている場合。
     デバッガが呼び出された状態に落ちています。C-d または、abort の番号を入力してプロンプトが">" になるまで復旧して下さい。復旧はエラーの原因が判明した後でするようにしてください。
  2. Lisp 自体がおかしくなっている場合。
     Lisp listener に C-c C-c を入力して、一度デバッガに落してから復旧します。C-c C-c を入力したときに、Lisp の Top level にもどるかどうかを問うメッセージが出る場合がありますが、"y" と答えてください。
  3. ミニバッファ上に以前取りやめた操作のメッセージが出る場合。
     ミニバッファの質問に答えずにマウスでカーソルを移動すると、そのときのメッセージだけが取り残される場合があります。これを復旧するには、C-x o コマンドを何度か使用してメッセージの取り残されたミニバッファを呼び出し、C-g コマンドで明示的に abort してください。

4. TDMT バッファを kill してしまった場合。

操作の誤りなどにより、TDMT バッファを kill してしまった場合にミニバッファに TDMT バッファを作りなおすかどうかを問うメッセージが出ることがあります。これには、"y" と答えることにより TDMT バッファが再度作りなおされます。また、強制的に TDMT バッファを作りなおしたい場合には、M-x tdmt-windows コマンドを使用してください。

第 5 章

環境設定

ここでは、システムをいつも同じ状態で使用するために必要な環境設定について説明します。

5.1 必要な環境

本ツールを使用するためには次のアプリケーションを使える環境が必要です。

1. mule-2.1
2. ILISP 5.6
3. Lucid Common Lisp 4.1 (non-dbcs)
4. TDMT-MULTI-4

5.2 .mule への登録

各ユーザーの.mule ファイルに例えば次のような登録が必要です。これにより、ILISP 環境がロードでき、さらに M-x run-tdmt をはじめとする TDMT 専用のコマンドが使えるようになります。

```
;;;
;;; For ILISP
;;;
(setq load-path (cons (expand-file-name "/usr/local/gnu/mule-2.1/lib/mule/site-lisp/ilisp/")
                      load-path))
(load "ilisp.emacs")

;;;
;;; For TDMT
;;;
(setq load-path (cons (expand-file-name "/usr/local/TDMT/tdmt-multi-4/emacs19-stuff/")
                      load-path))
(autoload 'tdmt-windows "setup-tdmt-windows" "TDMT-MULTI-4" t)
(autoload 'run-tdmt "setup-tdmt" "TDMT-MULTI-4" t)
```

5.3 セットアップファイルの作成

TDMT-MULTI システムはいくつかの言語間の翻訳を同時に扱えるようになっています。しかし、一般的には翻訳知識を作成する間は対象とする言語のみを扱えば十分です。また、翻訳知識を作成するための作業用のファイルやディレクトリは知識作成者の都合の良い場所におき、管理する必要があります。このため、あらかじめ作成者固有のセットアップファイルを用意して、その中で必要なモジュールおよびファイル、ディレクトリなどを指定し、自動的にシステムをセットアップできるようにします。

セットアップファイルの例を示します。

```
;;; -*- Mode: LISP; Syntax: Common-lisp; Package: COMMON-LISP-USER; Base: 10 -*-

;;;
;;; Load SETUP-INIT
;;;

(in-package :public)

(setq *jma-morph-dic-file*
      "/usr/local/TDMT/tdmt-multi-4/j-morph/dic/jma-morph-dic-jk.text")
(setq *jma-word-monogram-file*
      "/usr/local/TDMT/tdmt-multi-4/j-morph/ngram/word-monogram-jk.lisp")
(setq *jma-word-bigram-file*
      "/usr/local/TDMT/tdmt-multi-4/j-morph/ngram/word-bigram-jk.lisp")
(setq *transfer-j-k-dir* "/home/as45/dbkim/j-k/")


(load "/usr/local/TDMT/tdmt-multi-4/build/setup-init.lisp")


;;;
;;; Load other Sub Systems
;;;
(load-sub-system :j-morph)
(load-sub-system :transfer)
(load-sub-system :k-generation)


;;;
;;; Load dictionaries for each Sub System
;;;
(j-morph::j-load-all-data)
;;; check-mode ... exp) :if-warning-continue, :if-warning-exit, :no-check
(transfer::load-dictionaries :mode :j-k :check-mode :if-warning-continue)
(k-generation::load-data)
```

第 6 章

おわりに

本ツールはTDMTの翻訳知識を効率良く作成するために様々な機能を採り入れてバージョンアップを重ねてきました。開発当初は機能が少なく、操作性も不十分なため、利用者の方々にはいろいろと御迷惑をおかけしましたが、皆さんの御意見、御指摘のおかげでここまで性能向上ができました。今後も利用者の方々の御意見など参考にして操作性向上や機能向上のための改良を続けていきたいと思いますのでよろしくお願い致します。最後になりましたが、本ツールの開発および本報告書の作成にあたり有益な御意見を頂いた第三研究室の飯田室長、古瀬主任研究員、隅田主任研究員、金客員研究員、大井研究員、日本IRの徳橋さん、ニチメングラフィックスの栗原さん、コングレのフェリーさん、チェさんに感謝致します。

1994年12月28日

参考文献

[古瀬, 94] 古瀬, 隅田, 飯田：経験的知識を活用する変換主導型機械翻訳, 情報処理学会論文誌, Vol. 35, No. 3, pp.414-425, (1994).

[Furuse, 94] Furuse, O. and Iida, H.: Constituent Boundary Parsing for Example-Based Machine Translation, *Proc. of Coling '94*, pp.105-111, (1994).