

TR-IT-0196

音声認識システムと言語翻訳システムとの  
接続実験のための支援ツール

河井 淳 脇田 由実 古瀬 蔵

Jun KAWAI Yumi WAKITA Osamu FURUSE

1996年11月

概要

音声認識システムと言語翻訳システムとの接続実験を支援するためのツール群について説明する。カテゴリー N-gram による One Pass DP のフレーム同期型音声認識システム (HMnet-NGRAM) のための言語モデル作成、認識率評価のためのツールをはじめ、変換主導型多言語翻訳システム (TDMT-MULTI) におけるインターフェイス用のパッケージに関して説明する。また、音声言語データベース (SLDB) を利用したツールについても説明する。

エイ・ティ・アール音声翻訳通信研究所

ATR Interpreting Telecommunications Research Laboratories

©(株) エイ・ティ・アール音声翻訳通信研究所 1996

©1996 by ATR Interpreting Telecommunications Research Laboratories

# 目次

1	はじめに	1
2	音声認識システム (HMnet-NGRAM) について	2
2.1	概要	2
2.2	音声認識実験支援ツール	3
2.2.1	設定と使用方法	3
2.2.2	言語データベース関連	4
2.2.3	N-gram 言語モデル (N-gram) 関連	11
2.2.4	音声認識結果の評価	11
3	言語翻訳システム (TDMT-MULTI) について	12
3.1	概要	12
3.2	支援ツール	12
4	接続インターフェイス	14
4.1	音声認識と言語翻訳	14
4.2	言語翻訳と音声合成	14
4.3	サーバー型 TDMT のプロトタイプ	15
4.3.1	構成	15
4.3.2	入力仕様	16
4.3.3	出力仕様	17
4.3.4	サーバー型 TDMT プロトタイプを設計する際に考慮した点	18
5	その他	20
5.1	C-STAR II ワークショップのデモシステム	20
5.2	統合実験のためのマスター単語辞書	21
5.3	音声認識結果出力の標準フォーマット	28
6	おわりに	33

# 第 1 章

## はじめに

本報告書では主に音声認識システムと言語翻訳システムとの接続実験を支援するためのツール群について説明する。これは第三研究室において音声認識と言語翻訳の接続実験を行なうために作成したものであるが、システムに強く依存していないと考えられるものもある。また、1996年9月にATRで開催された C-STAR II ワークショップ[C-STAR II, 96]でのデモ「音声翻訳研究のための機能モジュール設計」で使用された TDMT インターフェイスのプロトタイプについても説明する。

## 第 2 章

### 音声認識システム (HMnet-NGRAM) について

#### 2.1 概要

現在 (1996 年 11 月) 第三研究室でメンテナンスを行なっている音声認識システムの仕様は次の通りである。これは第三研究室内部での予備実験などのために使用されるものであり、第一研究室の ATR-lattice や第四研究室の HMnet-LR とは独立のシステムである。(但し、HMnet 音響モデルは共通に使用できる。)

- 探索アルゴリズム  
N-best beam search (one pass DP, frame synchronous)
- 音響モデル  
Context dependent model (HMnet)
- 言語モデル  
Class/Word N-gram model (bi-gram, tri-gram)
- 出力形式  
N-best output (word sequences, strings)

出力フォーマットについては、従来から実験に使用していた第三研究室の独自フォーマットを基本としていたが、第一研究室の提唱する標準化フォーマット (5.3節参照) による出力も可能である。

---

## 2.2 音声認識実験支援ツール

本音声認識実験のために以下のような支援ツールがある。

- 言語データベース関連  
言語データベースから必要な条件に合うデータを取り出すためのツール。
- N-gram 言語モデル関連  
単語またはカテゴリの N-gram 言語モデルを作成するためのツール。
- 音声認識結果の評価  
音声認識結果を評価するためのツール。

上記の内、言語データベース関連のツールはシステム依存性は低いと考えられるので、他の実験のためにも再利用できるかもしれない。また、N-gram 言語モデル関連のツールは最終的なモデルのフォーマットが標準化されれば統一化できる可能性が高い。最後に音声認識結果の評価のツールも現状では音声認識出力標準フォーマットに対応していないが、今後対応したものに置き換えることを計画している。

なお、これらのツールでは単語（形態素）の ID とデータベースの整合を取るためにマスター単語辞書（5.2節）を参照する場合があるが、これは扱う単語（形態素）の体系の違いによって切替える必要がある。大部分のツールでは -dept4、-SLDB オプションにより通常の SLDB 体系に切り替わり、-dept3、-TDMT オプションにより第三研究室独自の TDMT 体系に切り替わるようになっている。

以下これらの支援ツールについて説明する。

### 2.2.1 設定と使用方法

以下に説明するツールのほとんどは現在このディレクトリにある。

```
/dept3/work3/KAWAI/DATA/tools
```

デフォルトの設定に問題なく、特にカスタマイズなどの必要がない場合は csh 上で次のコマンドを実行すればよい。または、これを個人の.cshrc に加えてもよい。（但し、alias の内容に問題がないかを事前に確認した方がよい。）

```
source /dept3/work3/KAWAI/DATA/tools/.alias
```

何らかのカスタマイズが必要な場合は、次のような方法が考えられる。

1. 変更が少しの場合  
.alias を個人用にコピーして編集する。
2. 変更が多い場合  
すべてコピーしてデフォルトを設定している次のファイルを編集する。  
db\_define.awk  
このファイルの詳細については後述する。このとき、.alias 中の tools のパスもコピー先へ変更しておく必要がある。

## 2.2.2 言語データベース関連

ここでは、言語データベースを扱うためのツールについて説明する。ここでいう言語データベースとは、音声言語データベース (SLDB) として音声データベースと関連付けられたものと、言語情報のみのデータベースとして独立に構築されたものを広く指す。これらの一部は仕様が文書にまとめられているが、そうでないものも含まれる [TR-IT-0003, 1993] [TR-IT-0009, 1993] [TR-IT-0053, 1994] [TR-IT-0056, 1994]。言語データベースにはテキストや形態素の情報があり、これらを利用して統計的な学習用のデータを作成したり、実験結果を評価するための正解ファイルを作成することができる。現在、データベースに一部不整合な箇所があるため完全に自動的にこれらの処理を行うことは難しい。しかし、大部分の処理は自動化可能であると考えられる。そこで、ある程度は自動的に処理しておいて、後から人手でチェックしながら修正するという方法が考えられる。

以下ではそのために用意したツールを説明する。

### • 会話 ID の処理

言語データベースから言語モデルの学習用セットを取り出したり、実験の評価用のセットを様々な条件で選定する必要がしばしば生じる。こういった場合に複雑な条件を満たすデータセットをデータベース上に見つけるのは非常に困難な作業となる。それはデータの存在と実験の目的に依存する要求を考慮する必要があるためである。これらの作業をできるだけ簡単に、しかも正確に行うために、まず会話 ID に注目してこれらを中心に会話を絞り込むことにした。具体的には会話 ID を第一フィールドに持つリストの形でセットを指定するファイルを作り、それらに対して集合和や集合積を行うことによって、目的の会話セットを得るものである。

リストファイルの例は次のディレクトリにある。

```
/dept3/work3/KAWAI/DATA/CONV.LIST
```

```
S1.list      testset S1
S2.list      testset S2

SL1.list     test SL1
SL2.list     test SL2
SL3.list     test SL3

SLDB.list    SLDB 618 会話 (1996 年 4 月 25 日現在)

TDMT.list    TDMT のトレーニングセット (1996 年 5 月 8 日現在)
TDMT_closed.list  1996 年 6 月 TDMT の評価用セット (closed)
TDMT_open.list  1996 年 6 月 TDMT の評価用セット (open)
```

以下のものは `get.info` で使用する特別なリストで第二フィールド以降の情報を利用して検索される。このファイルは第一研究室で作成されたものを利用している。なお、`conv_info_new.txt` はリンクなので注意が必要。

```
conv_info.txt
conv_info_new.txt -> /DB/SDB/ALL/INFO/all.ascii
```

なお、これらのリストファイルは `get.info` で使用されるものを除いて、第一フィールドのみを用いて処理されるので、第二フィールド以降の情報はユーザの必要に応じて活用することも無視することもできる。

以下にこれらの上記のリストを操作するコマンドを説明する。

---

`get_info CONV_INFO_FILE [-<conditions>] [-INF] [-help]`  
CONV\_INFO\_FILE から <conditions> の条件にあり行を選択する。

`get_info_new [-<conditions>] [-INF] [-help]`  
最新バージョンの CONV\_INFO\_FILE を参照する。

`get_info_old [-<conditions>] [-INF] [-help]`  
古いバージョンの CONV\_INFO\_FILE を参照する。

`mask_list AAA BBB [...]`  
AAA の各行を BBB および [...] でマスクする。

`and_list AAA BBB [...]`  
AAA と BBB および [...] の積集合を取る。  
行の先頭要素のみ出力する。  
コメント行は削除される。uniq して sort される。

`or_list AAA BBB [...]`  
AAA と BBB および [...] の和集合を取る。  
行の先頭要素のみ出力する。  
コメント行は削除される。uniq して sort される。

`del_list AAA BBB [...]`  
AAA から BBB および [...] の要素を取り除く。  
行の先頭要素のみ出力する。  
コメント行は削除される。uniq して sort される。

`com_out AAA [...]`  
AAA および [...] からコメント行を除きながら連結する。

使用例：

次の例では、WAV ファイルと JMOR ファイルがそろっていて、  
分類が B (担当者オリジナル) である会話のうち TDMT の  
テストセットに含まれるものを出力する。

```
% cd /dept3/work3/KAWAI/DATA/CONV_LIST/  
% get_info_new -wav -jmor -b | mask_list - TDMT.list  
TAS13005  
TAS13009  
TAS23001  
TAS33001  
TAS33011  
TBS23001  
TCS13021  
TCS13022  
THS33001  
TIS23001  
TKS33002
```

---

TOS23006

TRS33002

%

ここで、`get_info_new` にオプション `-inf` を指定すれば、`conv_info_new.list` の第二フィールド以降の情報も出力される。指定できる条件の詳細は `get_info_new -h` で確認できる。また、`mask_list` の第一引数の `-` はパイプからの標準入力を意味し、この `get_info_new` から送られた情報のうち `TDMT.list` に該当するものを抜きだしている。

次の例では、テストセット `SL1`、`SL2` のどちらかに含まれる会話のうち、`SLDB` と `TDMT` のセットに共通に含まれるものの件数を数える。

```
% or_list SL1.list SL2.list | and_list - SLDB.list TDMT.list | wc
      3      3      27
%
```

これにより、該当するファイルは 3 会話あることがわかる。



- テキスト抽出

言語データベースの中から入力の話 ID リストのものを対象として引数に与えられた条件に従ってテキストを取り出す。

#### get\_text

データベース (SLDB) から指定した条件に従ってテキスト情報を取り出す。

入力は第一フィールドが会話 ID のリストをファイルまたは標準入力を与える。

引数に指定できるオプションは以下のとおり。

```

[-(no_)j] [-(no_)e] [-(no_)k]           # Language
[-(no_)cus(=cli)] [-(no_)cle]           # Role: CUSTOMER(CLIENT), CLERK
[-a] [-b] [-c] [-z]                     # Speaker: A, B, C, Z
[-(no_)inf1]                             # information 1 (place of file)
[-(no_)inf2]                             # information 2 (condition
                                         of context)
[-(no_)sep_sent]                         # separating sentence
[-(no_)jt] [-(no_)et] [-(no_)kt]        # format text
[-(no_)not(_j|e|k)]                     # region of NOTES
[-(no_)rou(_j|e|k)]                     # region of ROUND parenthesis
[-(no_)sq(_j|e|k)]                      # region of SQUARE parenthesis
[-(no_)po(_j|e|k)]                      # character of POINT
[-(no_)sym(_j|e|k)]                     # character of some SYMBOL
[-all(_j|e|k)]                          # = -not -rou -sq -po -sym
[-(no_)lbl(_j|e|k)]                     # output label of ROLE
[-in_code(_j|e|k) file-coding]          # file coding ('internal'
                                         or 'euc-japan' etc.)
[-out_code(_j|e|k) file-coding]         # file coding ('internal'
                                         or 'euc-japan' etc.)
[-text_path text_path]                  # list of path of TEXT
                                         (delimiter=':')
[-add_text_path text_path]              # list of additional path of
                                         TEXT (delimiter=':')
[-def]                                   # default setting
[-mode]                                  # display current mode
[-help]                                   # help command
conv_id_list ...                          # files including IDs
                                         of conversations

```

#### 使用例:

```
% echo 'TAS12001' | get_text -j -a
```

すいません部屋の予約をしたいと思って電話したんですけど

わたくし一人だけです期間は八月の十日から十二日までです

一泊百ドルぐらいの部屋をお願いします

そうですかできればバスが付いての方がいいんですけど百ドルぐらいでそういった部屋は無

いですか

そうですかじゃあしかたありませんね百四十ドルのツインの方をお願いいたします

六時ごろには着くと思うんですけど

はい名前は鈴木和夫です

連絡先は今ニューヨークシティホテルに滞在しています六百二号室です

---

電話番号は二零三四四三一七零零です

カードはマスターカードで五二七九三九二零二四六九零零九八です

はい分かりましたそれと朝食はどうなってるんですかね

分かりましたじゃあよろしくお願ひします

はい分かりましたじゃ失礼します

%

- 形態素抽出

言語データベースの中から入力の話 ID リストのものを対象として引数に与えられた条件に従って形態素を取り出す。

`get_mor`

データベース (SLDB) から指定した条件に従って形態素情報を取り出す。

入力は第一フィールドが話 ID のリストをファイルまたは標準入力を与える。

引数に指定できるオプションは以下のとおり。

```

[-mor_sys morph_system]           # Switch System of Morphemes
[-dept3 -tdmt -dept4 -sldb]       # Switch System of Morphemes
[-LEX lexicon]                     # Lexicon
[-DEL_LEX del_lexicon]            # Lexicon of Deleting words
[-text_path text_path]            # list of path of TEXT
                                   (delimiter=':')
[-add_text_path text_path]        # list of additinal path
                                   of TEXT (delimiter=':')
[-mor_path mor_path]              # list of path of MOR
                                   (delimiter=':')
[-add_mor_path mor_path]          # list of additinal path
                                   of MOR (delimiter=':')
[-(no_)j] [-(no_)e] [-(no_)k]     # Language
[-(no_)cus(=cli)] [-(no_)cle]     # Role: CUSTOMER(CLIENT),
                                   CLERK
[-a] [-b] [-c] [-z]              # Speaker: A, B, C, Z
[-(no_)inf1]                       # information 1 (place of file)
[-(no_)inf2]                       # information 2 (condition
                                   of context)
[-(no_)sep_sent]                  # separating sentence
[-sn]                              # output sentence number
[-show_path]                       # show TEXT_, MOR_PATH_LIST
[-help]                            # help command
conv_id_list ...                   # files including IDs
                                   of conversations

```

使用例:

```

% echo 'TAS12001' | get_mor -j -a -dept4 | head -10
20||| 申込者: |||
20|0020|20|60| すいません | スミマセン | すみません | 感動詞 |||
20|0020|20|70| 。 ||。 | 記号 |||
20|0030|30|80| 部屋 | ヘヤ | 部屋 | 普通名詞 |||
20|0030|30|90| の | ノ | の | 連体助詞 |||
20|0030|40|100| 予約 | ヨヤク | 予約 | サ変名詞 |||
20|0030|40|110| を | ヲ | を | 格助詞 |||
20|0030|50|120| し | シ | する | 本動詞 | サ変 | 連用 ||
20|0030|50|130| た | タ | た | 助動詞 | 形容詞 | 語幹 ||
20|0030|50|140| い | イ | い | 語尾 | 形容詞 | 終止 ||
%

```

- デフォルト設定ファイルについて (db\_define.awk)  
データベース関連のツールは共通のデフォルト設定ファイルを使用している。以下に各設定の意味を説明する。

COCO Mule に付属しているコード変換ツールのパス。

UNKNOWN\_WORD["J"] 未知の日本語形態素を表す  
UNKNOWN\_WID["J"] 未知語の日本語形態素 ID

TEXT\_PATH\_LIST テキストファイルのパスリスト (: で区切られる)  
JMOR\_PATH\_LIST\_DEPT3 第三研体系の形態素 DB のパスリスト  
JMOR\_PATH\_LIST\_DEPT4 第四研体系の形態素 DB のパスリスト  
LEX\_DEPT3\_J 第三研体系のマスター辞書ファイル  
LEX\_DEPT4 第四研体系のマスター辞書ファイル  
DEL\_LEX\_DEPT3\_J 第三研体系の除く単語ファイル  
DEL\_LEX\_DEPT4 第四研体系の除く単語ファイル

MOR\_SYS 形態素体系のスイッチ (DEPT3 or DEPT4)

INPUT\_FILTER[LNG] 各言語 LNG の入力フィルタ  
OUTPUT\_FILTER[LNG] 各言語 LNG の出力フィルタ

ここに LNG はそれぞれの言語を表し、日本語 "J"、英語 "E"、韓国語 "K" を示す。(但し、現在では韓国語の結果は保証できない。)

JTEXT\_LBL[X] X で表される分類に対応した日本語ラベル  
ETEXT\_LBL[X] X で表される分類に対応した英語ラベル  
KTEXT\_LBL[X] X で表される分類に対応した韓国語ラベル(?)

ここに分類 X とは "A", "B", "C" で区分され、それぞれ申込者、担当者、通訳者を表している。また "" (空文字列) の時は前の発声と話者が交替せずに続いた場合の空ラベルを表す。

DEL\_J\_FILE 日本語テキストから条件により削除する正規表現ファイル  
DEL\_E\_FILE 英語テキストから条件により削除する正規表現ファイル  
DEL\_K\_FILE 韓国語テキストから条件により削除する正規表現ファイル

LNG\_J, LNG\_J, LNG\_J 対象言語の指定 (それぞれ "J", "E", "K")  
SPEAK\_A, SPEAK\_B, SPEAK\_C 対象話者の指定 (分類)  
ROLE\_CUSTOMER, ROLE\_CLERK 対象話者の指定 (役割)

SEP\_SENT 0: 発声単位、1: 文単位

ROUND\_J 等 0: 無効、1: 丸括弧部分を削除  
SQUARE\_J 等 0: 無効、1: 角括弧部分を削除  
POINT\_J 等 0: 無効、1: 句読点を削除  
SYMBOL\_J 0: 無効、1: 読みの注釈を削除

ALL\_J 等 0: 無視、1: すべての日本語を対象とする

---

LBL\_OUT\_J等 0: 無視、1: すべての日本語の話者ラベルを削除

INF1,INF2 0: 無効、1: 処理情報(条件など)も出力する

### 2.2.3 N-gram 言語モデル (N-gram) 関連

統計的言語モデルである N-gram を作成するためのツールと平均分岐数 (perplexity) を計測することによって静的に評価するツールがある。

cat\_entropy\_2.awk bigram 用

cat\_entropy\_3.awk trigram 用

これらはカテゴリの N-gram についてのエントロピー (パープレキシティ) を計算したり、モデルのカバー率を計測することができる。十分に整理されていないので、詳細は割愛する。

### 2.2.4 音声認識結果の評価

音声認識の結果を評価するためには、正解ファイルを用意しておき認識結果と比較する必要がある。正解ファイルを作成するツールと認識結果を評価するツールがある。

EvalResult.csh

これは第四研究室にて使用されていたものをベースに作られている。単語認識率を適合率 (precision)、再現率 (recall) を用いて表現している。十分に整理されていないので、詳細は割愛する。

## 第 3 章

### 言語翻訳システム (TDMT-MULTI) について

#### 3.1 概要

現在(96年11月)第三研究室で開発されている言語翻訳のシステムをここでは TDMT-MULTI と呼ぶ。現在 TDMT-MULTI は以下のようになっている[古瀬他, 95][Furuse, et al., 95]。

なお、性能の評価結果が報告されている[美馬他, 96][山本他, 96][Paul 他, 96]。

- 最新バージョン  
TDMT-MULTI-6 (fixed)  
TDMT-MULTI-6-dev (development)
- 翻訳対  
日本語 ↔ 英語  
日本語 ↔ 韓国語  
日本語 → ドイツ語
- 使用言語  
Common Lisp (Allegro Common Lisp 4.3)  
なお、十分にテストされていないが、Gnu Common Lisp 2.2 での動作も確認している。

#### 3.2 支援ツール

翻訳知識作成のために mule 上で動作するツールがある。これはエディタ上に TDMT のプロセスを立ち上げて動作する[河井, 94]。

なお、これを使用するためには以下の emacs-lisp コードを個人の .emacs などに設定する必要がある。

(上記のテクニカルレポートは若干古いので設定などは少し変わっているが、基本的な使用方法はほとんど変わっていない。)

---

```
;;;
;;; for LISP
;;;
(setq load-path
      (cons (expand-file-name
             "/usr/local/TDMT/tdmt-multi-6/emacs19-stuff/ilisp-5.7/")
            load-path))
(load "ilisp.emacs")
(if (or (equal (getenv "HOSTNAME") "atra03")
        (equal (getenv "HOSTNAME") "atra04")))
    (setq allegro-program "/data/atra03/acl4.3/bin/develop")
    (setq allegro-program "/data/as30/acl4.3/bin/develop")
  )

;;;
;;; For TDMT
;;;
(setq load-path (cons
                (expand-file-name
                 "/usr/local/TDMT/tdmt-multi-6/emacs19-stuff/")
                load-path))
(autoload 'tdmt-windows "setup-tdmt" "TDMT-MULTI" t)
(autoload 'run-tdmt "setup-tdmt" "TDMT-MULTI" t)
```

## 第 4 章

### 接続インターフェイス

#### 4.1 音声認識と言語翻訳

音声認識と言語翻訳を接続するために考慮しなければならないこととして、次のことが重要であると考えられる。

- 一般性  
渡される情報はできるだけ一般的なフォーマットに従う方が良い。
- 拡張性  
研究の状況などに応じて機能を拡張的に追加できる方が良い。
- 冗長性  
渡される情報に多少の重複があった方が良い。
- 頑健性  
予期しない情報に対しても対応できる方が良い。

上記のことを考慮した結果、次の二項目の仕様を明確化することによってそのガイドラインを具体的に示すこととなった。

- マスタ単語辞書仕様  
言語データベースの単語(ここでは形態素と同義)に対して、一意の識別番号(単語 ID)を対応付けした辞書の仕様。この辞書により、単語の内部情報(形態素情報)を圧縮して扱うことができる。また、データベースの更新に対して拡張的にマスタ単語辞書を更新できるようになっている。現在の置き場所は、

```
/dept1/work7/MASTER.LEX/MASTER.LEXICON
```

であり、同じディレクトリの下にフィックスのためにコピーされた JMOR や作成、更新のツールも置かれている。

- 音声認識出力フォーマット仕様  
音声認識システムの出力のフォーマットを定めたもの。主要な情報は特定のシステムに依存しない形で表現することによって、一定の互換性を確保し、また、新たに追加される項目は他に影響を与えないように拡張的に定義できる仕様となっている。

#### 4.2 言語翻訳と音声合成

現在のところ正式に明文化されたフォーマットは存在しないが、C-STAR II ワークショップのデモのために CHATR が必要な情報について若干検討され仮のフォーマットが設定された。音声認識出力フォーマットに準じたものになっている。



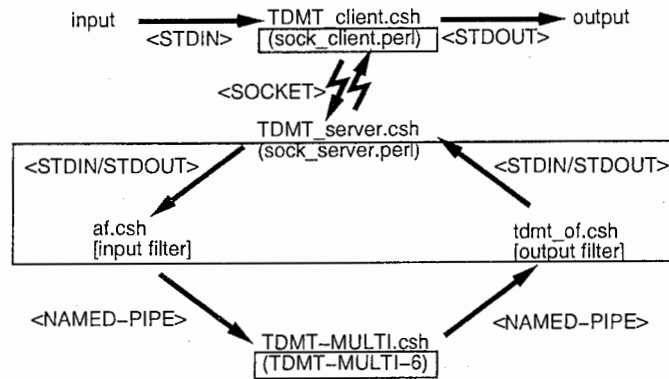


図 4.1: Server/Client system of TDMT-MULTI

### 4.3 サーバー型 TDMT のプロトタイプ

C-STAR II ワークショップでの「音声翻訳研究のための機能モジュール設計」のデモのためにサーバー型 TDMT のプロトタイプを作成した。これらは TDMT 本体にいくつかのスクリプト言語で書かれたツールを組み合わせる構成されている。本デモで使用されたものは次のディレクトリにある。

/data/as06/kawai/DEMO\_TEST/09SEP96

但し、TDMT の本体はここには置いていない。インターフェイスのための package とツール類が置かれている。これらはインストーラを使って個人の作業ディレクトリにコピーすることもできる。

その他、使用方法等は README を参照。

以下にこのプロトタイプシステムを説明する。

#### 4.3.1 構成

図 4.1 参照。

- TDMT 本体 (TDMT-MULTI.csh)  
Common Lisp で書かれており、TDMT-MULTI-6 のフィックスバージョンを使用。入力は日本語のみ、出力は英語、韓国語、ドイツ語に対応。インターフェイス用の package が追加されており、エラーハンドリングされたイベントループにより入力待ち状態に入る。
- 入力フィルタ (af.csh)  
af.csh から A-filter と呼ばれている形態素体系の変換フィルタが起動される。その本体は awk で書かれており、gawk 2.15 が使用された。
- 出力フィルタ (tdmf\_of.csh)  
tdmf\_of.csh から起動される i2jkg.perl は Mule の internal のコードから EUC や ISO-8859-1 コードに変換する perl スクリプトである。perl 4.019 で動作する。
- クライアント側モジュール (TDMT\_client.csh)  
TDMT\_client.csh からはサーバー側と通信するためのソケットプログラム (sock\_client.perl) が起動される。これは事実上、入力と出力のデータストリームを確保するだけである。なお、ソケットプログラムは perl 4.019 で動作する。
- サーバー側モジュール (TDMT\_server.csh)  
TDMT\_server.csh からは出力フィルタとソケットプログラム (sock\_server.perl) と入力フィルタをパイプで接続したプロセスが起動される。ソケットから受けとったデータを入力フィルタに渡し、出力フィルタから受けとったデータをソケットに返す。なお、ソケットプログラムは perl 4.019 で動作する。

### 4.3.2 入力仕様

入力は基本的に「音声認識出力標準フォーマット」に従うが、現在 TDMT では N-best 単語列（もしくは文字列）しか扱えないので N-best 単語列に関わる部分だけを処理する。それ以外の項目は無視する。（但し、ストリームを通して出力はされる。）その他、# で始まる行もコメントとして無視する。次の各行を入力として処理する。

#### 情報伝達用

UTTERANCE=...	発声 ID
NBEST=...	N-best の数（続く ORDER と wordids の組の数）
ORDER=...	N-best の順位
SENT=...	文字列（文字列渡しで使用。）
WORDS=...	単語列（/ で区切られる。現状では無視される。）
wordids=...	単語 ID 列（/ で区切られる。形態素渡しで使用。）

#### 制御用

TDMTCommand="0 (switch-mode :j-e)"	日英モード
TDMTCommand="0 (switch-mode :j-k)"	日韓モード
TDMTCommand="0 (switch-mode :j-g)"	日独モード
TDMTCommand="0 (translate \"こんにちは\")"	文字列翻訳 (EUC)

#### 詳細:

- UTTERANCE  
発声を特定するための ID。発声フィールドのマーカとして使用。
- NBEST  
音声認識が実際に出力した N-best の数。以下に続く ORDER 以降の組の数を指定している。今回は第一位候補のみを翻訳対象とすることになった。
- ORDER  
N-best 中の第何位の候補かを表す。
- SENT  
音声認識結果の文字列。文字列渡しのモードで使用される。（今回は未使用）
- WORDS  
音声認識結果の単語列の文字列表記。（今回は未使用）
- wordids  
音声認識結果の単語 ID 列。形態素渡しのモードで使用される。（今回はこのモードで使用）
- TDMTCommand  
TDMT へのコマンド。主に TDMT の制御用に使われる。フィールド内は  
"数字 リスブコマンド"

の形式をしており、switch-mode 関数で翻訳対を変更したり、translate 関数で直接文字列翻訳させることもできる。実行結果は TDMTResult として返される。なお、数字は何でもよく、制御側が識別できるコードを入れておけば、TDMTResult と対応を取ることもできる。

---

### 4.3.3 出力仕様

出力は、入力された行のすべてに TDMT からの処理結果を挿入したものである。それぞれの処理結果は対応する入力項目の直後に置かれるので、容易に対応を取ることができる。処理結果として次のものを出力している。

#### 情報伝達用

SOURCE_SENT="..."	入力文 (復元された表層文字列)
TARGET_SENT="..."	出力文 (画面表示用)
SYN_SENT="..."	出力文 (音声合成用)
tdmt_score=...	TDMT の尤度 (用例と入力の距離の合計)
tdmt_cands=...	TDMT の出力候補数 (このうち第一位のみ出力)
tdmt_morph="..."	形態素解析処理情報
tdmt_transfer="..."	変換処理情報
tdmt_source_tree="..."	原言語の木構造
tdmt_target_tree="..."	目的言語の木構造
tdmt_generation="..."	生成処理情報

#### 制御用

TDMTResult="..."	TDMTCommand に対する実行結果
TDMTError="..."	TDMT 内部で発声したエラー情報

#### 詳細:

- SOURCE\_SENT  
入力文字列。または音声認識から単語列で渡された入力文を文字列に復元したもの。
- TARGET\_SENT  
画面に表示するための翻訳結果文字列。
- SYN\_SENT  
音声合成処理のための翻訳結果文字列。現状ではほとんど TARGET\_SENT と同じだが、ドイツ語出力では違う。(将来、研究の余地あり。)
- tdmt\_score  
TDMT の尤度。第一位候補の木構造上での、用例と入力との距離の合計。
- tdmt\_cands  
TDMT の出力候補数。最終候補に残った数。現状では、この内の第一位のみが出力される対象となる。
- tdmt\_morph  
形態素解析処理情報。複数行出力される。最後に tdmt\_morph=:END が出力される。音声認識から単語列を受けとった場合でも数詞処理や形態素合成など、一部の処理が実行される。
- tdmt\_transfer  
変換処理情報。複数行出力される。最後に tdmt\_transfer=:END が出力される。

- 
- `tdmt_source_tree`  
原言語の木構造。フィールド内は  
|子ノード数@ 尤度@ ノードラベル  
の形で左から右へ木構造を深さ優先探索する順番でノードが並んでいる。
  - `tdmt_target_tree`  
目的言語の木構造。フィールド内は  
|子ノード数@ 尤度@ ノードラベル  
の形で左から右へ木構造を深さ優先探索する順番でノードが並んでいる。
  - `tdmt_generation`  
生成処理情報。複数行出力される。最後に `tdmt_generation=:END` が出力される。
  - `TDMTResult`  
`TDMTCommand` に対する返却値。フィールド内は  
”数字 返却値”  
であり、数字は `TDMTCommand` のものと対応する。
  - `TDMTError`  
`TDMT` の処理中にエラーが発生した場合に出力される。この場合、エラーの発生箇所以降に出力されるはずのものは出力されないで注意が必要。基本的には `TDMT` はエラーから回復され処理は続行できる。返されるメッセージの文字列は Lisp システムに依存したものとなる。

#### 4.3.4 サーバー型 TDMT プロトタイプを設計する際に考慮した点

本プロトタイプを設計するにあたって、考慮する必要のあった点について説明する。これは今後のデモシステムや実験システムの構築の際の参考になるかもしれない。

- 協調性  
音声認識、音声合成、制御といった他のモジュールと接続する必要があった。制御モジュールを介して情報をやりとりするが、情報の詳細な仕様は音声認識や音声合成の担当者レベルで決定した。これにより情報の制御と伝達内容をうまく分離することができた。
- 容易性  
処理に必要な入出力データと制御用のデータをすべて同様の枠組で扱った。これによりシステムのテストや改良が容易になった。処理速度やデータ衝突問題などに若干の課題があるかもしれない。
- 汎用性  
翻訳処理の本体は現在 Allegro Common Lisp という言語で動作しているが、基本的には Common Lisp であるので他の処理系に置き換えたり、C 言語などに移植される可能性もある。その場合の変更を皆無、もしくは最小にするために、インターフェイスの機構はできるだけ一般的なものにした方がよい。また、他言語による変換フィルタなどを前後に入れる必要もあった。そこで今回は「名前つきパイプ (named pipe)」という仕組みを利用した。これにより、Lisp 側からは単なるファイル読み書きにより入出力をとることができた。この他には標準入出力を利用する仕組みも考えられる。(今回はデバッグやモニタのために標準出力を利用した。)
- 柔軟性  
プロトタイプの段階なので、研究や実験のために構成を変える必要に迫られることが考えられる。そこで、できるだけ各部の役割を明確にして、必要に応じて再構成できるようにした。

---

- 耐久性

複雑な処理ではプログラムそのものや内部の辞書などに潜在的に存在する誤りのために、エラーを生じることがある。その場合にもシステム（モジュール）が停止しないようにする必要がある。今回は Lisp 内の interface package のイベントループに対してエラーハンドリングすることによってこれを防いでいる。もちろん、エラーが発生した場合は出力に若干の異常が生じるが、処理そのものはリセットされて続行される。

## 第 5 章

### その他

#### 5.1 C-STAR II ワークショップのデモシステム

1996年9月にATRで開催されたC-STAR IIワークショップ[C-STAR II, 96]では、翻訳処理に関わるものとしては二種類のデモが行われた。一つは多言語の翻訳処理に重点をおいたものであり、もう一つは音声認識、言語翻訳、音声合成までを統合することを目指したものであった。今回は時間の都合などの制約のためこれら二種類のデモが別々の枠組で構成されたが、今後は統一的に扱えるよう整理する必要がある。

- 多言語話し言葉翻訳システム  
日英、日韓の双方向と日独の多言語翻訳に TDMT 使用。  
入力は文字列。出力は表示および音声合成に CHATR 使用。  
CHATR には直接コマンドを送って実行している。
- 音声翻訳研究のための機能モジュール設計  
日本語の音声認識に ATRlattice、日本語から英語、韓国語、ドイツ語への翻訳に TDMT、音声合成に CHATR 使用。  
TDMT への入力は音声認識出力フォーマットを基本として  
TDMT 制御用のコマンドを追加したもの。  
TDMT の出力も同様の形式を踏襲して仮のフォーマットを使用。

---

## 5.2 統合実験のためのマスター単語辞書

これは、音声処理と言語処理を効率良く接続するために単語に一意のラベルを与える辞書である。現在は、

`/dept1/work7/MASTER.LEX/README`

に以下のドキュメントが置かれている。また、辞書の実体は、

`/dept1/work7/MASTER.LEX/MASTER.LEXICON`

作成元の形態素 DB のバックアップは、

`/dept1/work7/MASTER.LEX/JMOR/`

作成、更新のツールは、

`/dept1/work7/MASTER.LEX/Tools/`

の下にある。

なお、第三研究室内部で使用している形態素体系によるマスター単語辞書も存在する。これも全く同一の仕様を採用している。現在、その実体は

`/dept3/work3/KAWAI/DATA/DIC_TDMT/MASTER.LEXICON.DEPT3`

にある。

---

## マスター単語辞書について

第三研究室 河井 淳

1996年3月12日

### 1. 目的

マスター単語辞書は、モジュール間インタフェースにおける単語情報授受における整合性を確保することを主な目的とする。ここでは、SLDB、SDB、LDBで付与されている形態素属性の組に対し一意の単語ラベルを与えることで、これを実現する。このマスター辞書は、各モジュールの個別辞書を作成したり、モジュール間インタフェースの単語を指定したり、モジュール評価用の正解ファイルを作成するために用いられることを想定している。従って、発音記号(識別番号)、品詞クラス、アクセント位置等のモジュールに依存する情報は各モジュールごとに個別辞書に別途定義し、基本的に単語ラベルによってマスター辞書とリンクされるようにする。

### 2. マスター単語辞書に含める項目

マスター辞書には以下の項目を含め、フォーマットも定義する。(詳細は5.2)

- 2.1. 単語ラベル : 単語 ID としての役割を持つ。
- 2.2. 単語ボディー : JMOR の表記形・カナ表記・標準形・品詞・活用1・活用2・コメント(現在は音便の種別が入っている)

\*但し、将来、もし他言語に拡張される場合には、各言語ごとに単語ボディーを定義することとする。

### 3. マスター辞書の管理

#### 3.1 単語ラベル

単語の追加に対しては、単語ラベルの追加を行なう。  
単語の修正に対しては、修正後単語を新規に追加する。  
単語の削除は行なわない。

#### 3.2 マスター単語辞書の作成更新

マスター単語辞書の作成、更新を行なう管理者を置く。  
管理者はマスター辞書の更新を行ない、更新された辞書を公開する。  
マスター辞書は常にDBまたは、特定のDBのコピーと対応していなければならない。「マスター単語辞書作成更新ツール」により自動的にDBから作成、更新することにより、整合性を確保し運用負荷を軽減する。  
また、新しいバージョンの辞書は通常以前のバージョンの辞書を含むように作成されることとする。

### 4. マスター単語辞書の利用上の注意



#### 4.1 複製について

混乱を避けるため、一般利用者は可能な限り複写を避け、オリジナルは  
もちろん複写の編集も基本的に禁止する。

万一、編集する必要が生じた場合はコメント行にその履歴を明示し、  
さらにファイル名を変える等により単なる複写でないことを明かに  
する必要がある。

#### 4.2 内容について

辞書に登録されている内容は、修正等により、既に使用されていないもの  
を含んでいる可能性がある。従って、どの単語を使用すべきかは各モジュ  
ールごとに選択し、定義する必要がある。また、辞書項目が「表記形」や  
「標準形」および単語ラベル等で常にソーティングされていることは保障  
されない。

#### 4.3 情報のリンク

マスター単語辞書と各モジュールの個別辞書とのリンクは、基本的には  
個別辞書に単語ラベルを記述することにより実現する。万一、何らかの  
理由でこれが困難な場合でも、容易に単語ラベルを特定可能な情報を用い  
ることでモジュール間の情報授受を行なうものとする。  
また、トラブルを避けるためにも個別辞書には対応するマスター単語辞書  
のバージョンを常に記述しておくことが推奨される。

### 5. マスター単語辞書仕様

#### 5.1 単語の単位

単語の単位は、データベースの形態素解析結果 (JMOR 等) の単位と同一とする。  
単語辞書は、形態素解析結果ファイルにあらわれる形態素情報を持つすべ  
ての行 (すなわち、発話ラベルの行を除くすべての行) に対応する単語を  
含む。すなわち、句読点、記号類や、単語中に間投詞や言い直しを含むも  
のにも、すべて単語ラベルを与える。

#### 5.2 マスター単語辞書の形式

##### 1) バージョン情報

辞書更新ツールにより、辞書のバージョンおよび更新日時がコメント行  
に記録される。バージョンは3桁の独立した整数で表現される。

```
[version] := [dic_name] [root].[db].[table] [language]
[dic_name] := 辞書種別を示す。(現在は M-lex のみ)
[root]      := 初期作成の起源を示す。
[db]       := データベース (JMOR 等) に関する更新を示す。
[table]    := 他の辞書からの拡張による更新を示す。
[language] := 言語種別を示す。(現在は jp のみ)
```

(具体例)

```
##Version=M-lex 1.1.2 jp
```

##### 2) 単語情報

単語情報は、単語ラベルと単語ボディーからなる。

- 1 行に 1 単語を記述する。
- "#" に後続する文字列はコメントで、プログラムでは無視する。
- コメント行、空行を除く各行のフォーマットは以下の通りとする。

空白 \* 単語ラベル 空白 + 単語ボディー [空白 \* "#" コメント]

ただし、

空白 \* = 空白の 0 個以上の繰り返し

空白 + = 空白の 1 個以上の繰り返し

空白 = 1 byte スペースまたはタブ (2 byte のスペースは含まない)

単語ラベル = アスキー文字列

単語ボディー = 形態素情報の a から g のフィールド末尾を "|" で区切ったもの

[ ] 内は省略可能。

単語ラベルおよび形態素情報は、空白を含まないものとする。形態素情報の各項は空でもよい。ただし、区切り記号は省略してはならない。

#### • 単語ラベル

空白を含まないアスキー文字列とする。但し、便宜上整数値を与える。(整合性のため上位桁を 0 で埋めない。000 や 0013 は使用しない。) DB 上の単語以外に幾つかの疑似単語を登録する。

##### a) 疑似単語 (単語ラベル: 0-99)

これらは単語ボディーのフォーマットを別に定義する。

例えば、以下のようなアスキー文字列とする。

単語ラベル	単語ボディー	意味
0	PAUSE	# ポーズ
1	SENT-START	# 文の開始
2	SENT-END	# 文の終了
3	PU-START	# ポーズ節の開始
4	PU-END	# ポーズ節の終了
5	UTT-START	# 発声の開始
6	UTT-END	# 発声の終了
7	TURN-START	# ターンの開始
8	TURN-END	# ターンの終了

9 - 99 \*疑似単語用ラベルとして予約

##### b) 一般単語 (単語ラベル: 100-)

SLDB、SDB、LDB で付与された形態素はすべて一般単語として登録される。従って、DB 上の「記号」や「その他」の品詞が付いた単語も登録される。また、単語ボディーの内容が異なるものはすべて別単語として登録される。

- ・単語ボディー
  - a. 表記形 (形態素解析結果の第 5 フィールド)
  - b. カナ表記 ( " 第 6 フィールド)
  - c. 標準形 ( " 第 7 フィールド)
  - d. 品詞 ( " 第 8 フィールド)
  - e. 活用 1 ( " 第 9 フィールド)
  - f. 活用 2 ( " 第 10 フィールド)
  - g. 音便 ( " 第 11 フィールド)

これらがすべてが一致する単語を同一単語として一つの単語ラベルを与える。これらの少なくとも 1 つが異なるものは別単語として扱い、別の単語ラベルを与える。a - g は、DB 上の形態素解析結果の対応するフィールドと同じものである。

### 3) コメント行

- ・"#"ではじまる行はコメントで、プログラムでは無視する。
- ・空行（改行のみの行で、スペースやタブも含んではならない）は無視する。

## 6. ファイル内構成

ファイル内の構成は次のとおりとし、通常は以前のバージョンを含むように更新されていく。

----- file begin -----

固定ヘッド (バージョンによらず常に固定される情報を含むコメント群)

新バージョンヘッド (新バージョンの固有情報を含むコメント群)

新バージョン追加単語 (新バージョンで追加された単語群)

旧バージョンヘッド (旧バージョンの固有情報を含むコメント群)

旧バージョン追加単語 (旧バージョンで追加された単語群)

さらに旧バージョンヘッド

さらに旧バージョン追加単語

... 省略 ...

----- file end -----

## 7. 具体例

以下にマスター単語辞書の具体例を挙げる。これはフォーマットを確認するためのものなので、内容は実際の辞書情報と異なることがある。なお、コメントはメンテナンスやその他の利用のための機械処理が考慮されている。特に'##xxxxx=yyyyy'形式のものは機械処理のためのものである。

機械処理のためのコメント群：

Language	言語種別
Version	バージョン情報
Date	日付情報
Total_Number	全単語数(疑似単語数 + 一般単語数)
Pseudo_Number	疑似単語数
General_Number	一般単語数
MOR	参照形態素ディレクトリ
Lexicon	マスター単語辞書のパス名
Tools	更新ツールのパス名
Updater	更新者のアカウント名

----- begin -----

##### マスター単語辞書 (日本語) #####

# 注意:

# 利用者の方はこの辞書を編集しないでください。万一、変更の必要のある場合は編集日付等をコメント行に加え、ファイル名を変えて明示して下さい。

#

##### Master Lexicon (Japanese) #####

# CAUTION:

# A USER should not edit this file. If you need to modify by any chance, please add the comment included the date and etc., moreover clarify your modification by changing the file name.

#

##Language=Japanese

#

##END-OF-FIXED-HEADER#####

##Version=M-lex 1.01.00 jp

##Date=Fri Mar 8 14:48:11 JST 1996

##Total\_Number=8072

##Pseudo\_Number=9

##General\_Number=8063

##MOR=/dept1/work7/MASTER\_LEX/JMOR

##Lexicon=/dept1/work7/MASTER\_LEX/MASTER\_LEXICON

##Tools=/dept1/work7/MASTER\_LEX/Tools/BIN.HP-UX/update\_dic

##Updater=yama

##Updater's Comment

## β 第一版です

10000 いらっしゃいませ | イラっしゃイマセ | いらっしゃいませ | 感動詞 ||||

10001 。 ||。 | 記号 ||||

10002 こんにちは | コンニチハ | こんにちは | 感動詞 ||||

10003 、 ||、 | 記号 ||||

10004 [|| [| 記号 ||||

10005 あ | ア | あ | 間投詞 ||||

10006 ] ||] | 記号 ||||

10007 一 | ヒト | 一 | 数詞 ||||

10008 つ | ツ | つ | 接尾辞 ||||

10009 部屋 | へや | 部屋 | 普通名詞 ||||

10010 が | ガ | が | 格助詞 ||||

---

10011 欲し | ホシ | 欲し | 形容詞 | 形容詞 | 語幹 || ~  
10012 い | イ | い | 語尾 | 形容詞 | 連体 ||  
10013 ん | ノ | の | 準体助詞 ||||

... 省略 ...

----- end -----

#### 8. 各研究室代表者

マスター単語辞書の作成、更新、運用の方針は次の代表者により協議の結果決定された。(1996年2月)

第一研究室 政瀧, 清水  
第二研究室 Campbell  
第三研究室 河井, 脇田  
第四研究室 竹沢

(敬称を略させていただきました。)

以上

---

### 5.3 音声認識結果出力の標準フォーマット

第一研究室が提唱する音声認識出力の標準フォーマットの仕様を一覧にまとめたドキュメントがある。これは音声認識出力を Lattice(word graph)、N-best word/string sequence のいずれで出力する場合にも対応した統一した規格を目指して作成されたものである。

これにより、処理アルゴリズムや出力可能な形式の異なる様々な音声認識システムの出力を统一的に扱うことができ、性能比較やインターフェイスのマッチングを容易にすることができる。

現在、ドキュメントは

`/home/atrh29/shimizu/html/sprec/output.format.v11-2.txt`  
に置かれている。

以下にこのドキュメントを添付する。

Jun. 7 1996

[Syntax]

note: ( ) indicate that terms within may appear in any order

(要素の出現順序が任意)

{ } indicate that terms within are optional

(要素は省略可能)

1. Lattice

```
lattice_file      = header size_def terms
header            = general_header | utterance_header {header}
general_header    = { comment {\n} VERSION {\n} base {\n} lmname {\n}
                  lmscale {\n} wdpalnty {\n} }
utterance_header  = { UTTEANCE {\n} SUBLAT {\n} amname {\n} utime {\n}
                  abstime {\n} cputime {\n}}
size_def          = ( NODES LINKS ) \n
terms             = node_def | link_def {terms}
node_def          = I { time WORD wordid L var div fsanode acoustic } \n
link_def          = J ( START END { WORD wordid var div fsanode acoustic
                  ngram language score } ) \n
```

2. N-best

```
nbest_file        = header nbest_size_def nbest_terms
header            = general_header | utterance_header {header}
general_header    = { comment {\n} VERSION {\n} base {\n} lmname {\n}
                  lmscale {\n} wdpalnty {\n} }
utterance_header  = { UTTEANCE {\n} SUBLAT {\n} amname {\n} utime {\n}
                  abstime {\n} cputime {\n}}
nbest_size_def    = NBEST \n
nbest_terms       = nbest_terms1 | nbest_terms2
nbest_terms1      = NBEST ORDER SENT { wordids vars divs fsanodes
                  acoustic ngram language score } \n
nbest_terms2      = NBEST ORDER WORDS { wordids vars divs fsanodes
                  acoustic ngram language score } \n
```

[Field Types]

The currently defined fields are as follows:-

note: o -> optional(任意), c -> compulsory(必須)  
 + -> field defined in "HTK standard lattice format (SLF)"  
 (HTK 2.0 の標準ラティスフォーマットで定義されている  
 フィールド)

Field	abbr	o c	+	description
-------	------	-----	---	-------------

---

## 1. Header fields

(general recognition condition)

comment=%s		o		Comment [コメント]
VERSION=%s	V	o	+	Program version and machine environment [プログラムバージョンとマシン環境]
base=%f		o	+	LogBase for likelihoods (0.0 not logs, default base e)
lmname=%s		o	+	Name of language model [言語モデル名]
lmscale=%f		o	+	Scaling factor for language model [言語モデルの混合比]
wdpenalty=%f		o	+	Word insertion penalty [単語挿入ペナルティ]

(information for each utterance)

UTTERANCE=%s	U	o	+	Utterance identifier [発声識別子]
SUBLAT=%s	S	o	+	Sub-lattice name [Sub-lattice 名]
amname=%s		o		Name of acoustic model [音響モデル名]
utime=%f		o		Length of utterance [発声時間長]
abstime=%f		o		Absolute time start of utterance [発声開始の絶対時間]
cputime=%f		o		Cpu-time [CPU 時間]

## 2. Lattice

Lattice Size fields

NODES=%d	N	c	+	Number of nodes in lattice [ノード数]
LINKS=%d	L	c	+	Number of links in lattice



[リンク数]

Node fields

I=%d		c +	Node identifier [ノード番号]
time=%f	t	o +	Time from start of utterance [単語の開始相対時間]
WORD=%s	W	wc +	Word (If lattice labels nodes rather than links) [単語表記]
wordid=%s		wo	Word identifier [単語識別子]
L=%s		wc +	Substitute named sub-lattice for this node [Sub-lattice 名]
var=%d	v	wo +	Pronunciation variant number [発音種別]
fsanode=%d		wo	FSA node number [FSA ノード番号]

Link fields

J=%d		c +	Link identifier [リンク番号]
START=%d	S	c +	Start node number of the link [発声開始ノード]
END=%d	E	c +	End node number of the link [発声終了ノード]
WORD=%s	W	wc +	Word (If lattice labels links rather than nodes) [単語表記]
wordid=%s		wo	Word identifier [単語識別子]
var=%d	v	wo +	Pronunciation variant number [発音種別]
div=%s	d	wo +	Segmentation [セグメント列]
fsanode=%d		wo	FSA node number [FSA ノード番号]
acoustic=%f	a	wo +	Acoustic likelihood of link [音響尤度]
ngram=%f	n	o +	N-gram likelihood of link [リンクの N-gram 尤度]
language=%f	l	o +	General language model likelihood of link [言語尤度]
score=%f		wo	Total score [総スコア]

---

### 3. N-best

#### N-best Size fields

NBEST=%d	c	Size N [Nのサイズ]
----------	---	-------------------

#### N-best candidate fields

ORDER=%d	c	Order $n \leq nbest$ [順位 n]
SENT=%s	wc	N-th candidate sentence [n位の漢字仮名文字列]
WORDS=%s	wc	N-th candidate word sequence [n位の単語列]
wordids=%s	o	N-th candidate word identifiers [n位の単語識別子列]
vars=%d	o	N-th candidate pronunciation variant numbers [n位の発音種別列]
divs=%s	o	Segmentations [n位のセグメント列]
fsanodes=%d	o	FSA node numbers [FSA ノード番号列]
acoustic=%f	a	o + Acoustic likelihood of utterance [音響尤度]
ngram=%f	n	o + N-gram likelihood of utterance [リンクのN-gram尤度]
language=%f	l	o + General language model likelihood of utterance [言語尤度]
score=%f	o	Total score [総スコア]

### 4. Comment field

Allow comment fields anywhere in the file. Comment field is identified by a unique character "#" in the first column. Blank lines are ignored.

#### Note.

- 1) Either "SENT" field or "WORDS" field is compulsory (wc).  
"abbr" is a single character abbreviation for the field name.
- 2) Writing word sequence are allowed for "WORD, wordid, var, div" fields as well as "WORDS, wordids, vars, divs" fields.  
(Use "/"(slash) as a word boundary)

-----end of document-----

## 第 6 章

### おわりに

本報告書では主に音声認識と言語翻訳の接続実験のための支援ツールについて説明した。全く独立に設計されたモジュールを接続して実験するためには様々な問題を解決する必要があるが、中でも各モジュールの機能をうまく引き出すことは難しい。特にモジュール間のインターフェイスについては今後も一層の相互協力を重ねる必要があると思われる。

### 謝辞

最後になりましたが、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および御協力を頂いた第三研究室の飯田室長、隅田主任研究員、Paul 滞在研究員、ニチメングラフィックスの栗原氏、また、第一研究室の清水主任研究員、政瀧研究員、山本研究技術員、Singer 滞在研究員、第二研究室のCampbell 研究員、Hwang 滞在研究員、第四研究室の竹沢主任研究員、SET の林氏、TSG の Reaves 氏、伴氏に感謝致します。

1996 年 11 月 29 日

## 参考文献

- [TR-IT-0003, 1993] 浦谷, 竹沢, 田代, 衛藤: 音声言語データベースのための日本語形態素情報と表記の体系, *ATR Technical Report, TR-IT-0003*, Jul 1993.
- [TR-IT-0009, 1993] 浦谷, 田代, 山田, 松本: 音声言語データベースにおける日本語形態素解析マニュアル *ATR Technical Report, TR-IT-0009*, Sep 1993.
- [TR-IT-0053, 1994] 浦谷, 田代, 森田: 音声言語データベースにおける日本語形態素解析マニュアルの補遺 *ATR Technical Report, TR-IT-0053*, May 1994.
- [TR-IT-0056, 1994] 浦谷, 竹沢, 松尾, 森田: 音声言語データベースの構成, *ATR Technical Report, TR-IT-0056*, May 1994.
- [古瀬他, 95] 古瀬, 赤峯, 河井, 金, 飯田: "経験的な言語知識を利用する対話翻訳機構—日英・日韓の対話翻訳システム—", 言語処理学会第1回年次大会, 1995.3
- [Furuse, et al., 95] Furuse, Kawai, Iida, Akamine, Kim: "Multi-lingual Spoken-Language Translation Utilizing Translation Examples", 3rd Natural Language Processing Pacific Rim Symposium (NLPRS-95), 1995.12.
- [美馬他, 96] 美馬, 古瀬: 日英間変換主導翻訳の中間時評価, *ATR Technical Report, TR-IT-0189*, エイ・ティ・アール音声翻訳通信研究所, 1996.
- [山本他, 96] 山本, 古瀬: 日韓間変換主導翻訳の中間時評価, *ATR Technical Report, TR-IT-0190*, エイ・ティ・アール音声翻訳通信研究所, 1996.
- [Paul 他, 96] Paul, 伝, 古瀬: 日独変換主導翻訳の中間時評価, *ATR Technical Report, TR-IT-0191*, エイ・ティ・アール音声翻訳通信研究所, 1996.
- [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).
- [河井, 94] 河井: "TDMT 翻訳知識作成ツールの利用方法", *TR-IT-0088*, 1994.12.
- [C-STAR II, 96] C-STAR II 96 ATR International Workshop on Speech Translation, *Proc. of C-STAR II 96 Workshop*, (1996).