

TR-IT-0349

ATR-MATRIX データフォーマット仕様書

松田 猛† 西野 敦士 横尾 昭男
Takeshi MATSUDA† Atsushi NISHINO Akio YOKOO

2000.2.2

内容概要

音声翻訳システム (ATR-MATRIX) は様々なフォーマットのデータの制御を行う。ここでは、MATRIX 制御プログラムで使用されるデータのフォーマットを説明する。

エイ・ティ・アール音声翻訳通信研究所

ATR Interpreting Telecommunications Research Laboratories

©エイ・ティ・アール音声翻訳通信研究所 2000

©2000 by ATR Interpreting Telecommunications Research Laboratories

目次

本ドキュメントについて	3
1 ATR-MATRIX 上でのデータの種類	4
データの種類	4
ATR-MATRIX で使用するデータ	4
2 コントローラー間通信パケットデータ	5
フォーマット	5
送信先アドレス	5
送信元アドレス	5
メッセージ	6
データサイズ	6
データ	6
3 音声認識結果データ	7
4 言語翻訳入力データ	10
5 言語翻訳結果データ	12
6 音声合成用データ	15
言語フィルター (GFILTER) を使った音声合成用データの作成	15
kan2rom を使った音声合成用データの作成	17
CHATR 入力用音声合成データのフォーマット	17
7 音声認識結果表示用データ	18
8 言語翻訳結果表示用データ	19
9 対コミュニケーションサーバー通信データ	20
10 GUI コマンド	21
索引	22

本ドキュメントについて

ATR-MATRIX は日本語から英語、ドイツ語、韓国語、中国語へ、また、英語から日本語と言った異なる言語間での人間同士のコミュニケーションを可能にする音声翻訳統合システムです。ATR 音声翻訳通信研究所で開発された基礎技術「音声認識」、「言語翻訳」、「音声合成」の機能を統合し、デモンストレーション、研究開発支援などを目的として開発されました。

本ドキュメントは「ATR-MATRIX version 2.5.0」のソフトウェア内で使用されるモジュール間、コントローラー間、システム間のすべての授受「データ」に関する仕様書、および解説書です。このドキュメントは ATR-MATRIX の保守、管理、修正、機能追加などで、ATR-MATRIX ソフトウェアを更新するに当たり有効な資料となるものとして作成されました。

このドキュメントでは ATR-MATRIX の中でもテクニカルサポートグループが開発したシステム全体の流れを制御管理するシステム ATR-MATRIX version2.5.0 上で使用されるのデータフォーマットについて詳細に解説しております。

1 ATR-MATRIX 上でのデータの種類

データの種類

データの種類としては、

- コントローラー間通信パケットデータ
- 音声認識結果データ
- 言語翻訳結果データ
- 音声合成用データ
- 音声認識結果表示用データ
- 言語翻訳結果表示用データ
- 対コミュニケーションサーバー通信データ
- GUI コマンド

が存在します。

ATR-MATRIX で使用するデータ

ATR-MATRIX には 7 つのコントローラーと呼ばれる機能ごとのプログラムが存在します。(詳細は ATR-MATRIX version2.5.0 Software 仕様書参照) このコントローラー間で使用するデータや、各研究室で開発されたリソースとの通信をおこなうためのデータ、画面表示用に抽出したデータなどのデータが存在します。以後の章では、すべてのデータについて概略と使用される個所、およびフォーマットの説明をおこないます。

2 コントローラー間通信パケットデータ

7つのコントローラー（ATR-MATRIX version2.5.0 Software 仕様書参照）間では、仕様に基づいた共通のデータフォーマットを使用しています。これを通信パケットと言います。この通信パケットデータが実際にはどのようにやり取りが行われるかについては、松田 猛他著「ATR-MATRIX 通信パケットメッセージ仕様書」（TR-IT-0352）を参照してください。

フォーマット

送信先アドレス	送信元アドレス	メッセージ	データサイズ	データ
32byte	32byte	32byte	10byte	可変長

送信先アドレス

この通信パケットの送り先が記述されています。設定可能値は、

- SipMainController
- SipSprecController
- SipTdmtController
- SipChatrController
- SipGuiController
- SipConnectController
- SipSoundController

の7種類です。この項目は固定長で、文字列は前詰めで設定されており、空白部分には Fill 文字として、「.」（半角ピリオド）が入っています。

例) "SipMainController....."

送信元アドレス

この通信パケットの送り元が記述されています。設定可能値、Fill 文字は「●送信先アドレス」と同じです。

メッセージ

メッセージは通信パケットの意味を表します。コントローラーへの制御メッセージや、データの種類等が記述されています。また、メッセージが同じでも受け取ったコントローラーによって処理内容は違います。設定可能値は

- hello
- byebye
- initialize
- start
- result
- end
- exit
- error
- log
- recog
- trans
- csTrans
- host
- connect
- accept
- ready
- button
- awake
- sleep
- check
- language
- speaker

の全22種類です。この項目も固定長で、文字列は前詰めで設定されており、空白部分には Fill 文字として、「.」（半角ピリオド）が入っています。

データサイズ

パケット後部の「データ」のデータ長が数字で記述されています。この項目は固定長で、文字列は前詰めで設定されており、空白部分には Fill 文字として、「*」（半角アスタリスク）が入っています。

例) "*****10"

データ

通信パケットのデータ部分です。この項目に関しては存在しない場合もあります。(制御用通信パケットの場合など) 可変長で、Fill 文字もありません。この部分のデータサイズが通信パケットの「データサイズ」に記述されています。

3 音声認識結果データ

音声認識結果データは音声認識モジュール SPREC の出力データです。この出力データには詳細な音声認識結果の情報が含まれています。このデータは加工され GUI 表示用データや、言語翻訳入力データになります。

例)

```
VERSION=ReleaseATRLATTICE 06a02 Linux
frameshift=0.010000
base=1.000100
lmname=Lexicon(/dept1/work1/MatrixJ/V3/lmodel/RECOG_LEX.vn1000),Ngram(/dept1/work1/MatrixJ/V3/lmodel/vngram.equal.bin)
lmscale=12.000000
wdpenalty=10000.000000

comment=FILE-END
UTTERANCE=1
amname=[/dept1/work1/MatrixJ/V3/amodel/AM.M.803.5.bin],/dept1/work1/MatrixJ/V3/amodel/AM.F.803.5.
bin
utime=3.130000
abstime=2.580000
cputime=3.980000
NBEST=1
ORDER=1 WORDS=UTT-START/SENT-START/はい/SENT-END/SENT-START/ニューワシントンホテル/で+ございます
/SENT-END/UTT-END wordids=5/1/10031/2/1/10525/10002+10003/2/6 vars=1/1/1/1/1/1/1/1
divs=-,0.130000,1177984.000000/h,0.060000,221376.000000+a,0.080000,93040.000000+i,0.180000,19005
8.000000+-,0.020000,104075.000000//n,0.090000,258760.000000+j,0.070000,136852.000000+u,0.040000,
60900.000000+u,0.030000,
:
:
times=0.000000/0.130000/0.130000/0.470000/0.470000/0.470000/1.630000/3.130000/3.130000
score=4939555.000000 acoustic=6764755.000000 ngram=-1345200.000000 language=-480000.000000

.....キーワード
```

上記の音声認識結果データである SPREC の出力データの内、キーワードとなっている部分は最低限必要です。他の部分はなくてもかまいません。では、そのキーワードの部分がどのように使われているかを説明します。

■ **UTTERANCE=N**

データのシリアル番号と同時に、必要なデータの始まりを意味します。このデータを使用するあらゆるコントローラーで必要となっています。

■ **amname=[X],Y,Z**

話者選択の結果です。音声合成時に「男性話者」を使用するか、「女性話者」を使用するかという選択時に使用します。大括弧で括られている部分が、選択された話者です。男性女性の判定は、選択された話者の文字列に「AM.M」「M00」の文字列が含まれていれば「男性話者」、「AM.F」「F00」の文字列が含まれていれば「女性話者」、どの文字列も含まれていなければ「男性話者」となっています。

■ ● **NBEST=N**

N が複数の場合複数の認識結果が存在すると判断します。認識結果が複数存在する場合最も尤度の高いもの（一番上位に記述されているもの）のみを取得します。様々なデータ抽出時に必要なキーワードです。

■ **ORDER=N**

この 1 行が音声認識結果の中身になります。この行の中にまた、キーワードが含まれており、それぞれの意味を持ちます。

■ **ORDER=N**

認識結果が複数ある場合、何番目の認識結果の内容かを意味します。単数の場合は N=1 となります。

■ **WORDS=UTT—START.....UTT—END**

「かな漢字混じり文字列」認識結果です。

複文認識の場合はデータ部分の SENT—START.....SENT—END が複数存在します。例のデータを参照してみてください。この部分は主に GUI 表示用に使われます。

■ **wordids=5/...../6**

「TDMT 体系形態素 ID」の認識結果です。

複文認識の場合はデータ部分の 1/...../2 が複数存在します。例のデータを参照してみてください。この部分は主に翻訳用データとして使われます。

■ `divs=.....`

発音記号データです。

発音記号表示用にアルファベットを抽出して使用します。

4 言語翻訳入力データ

■言語翻訳入力データは、先に説明してあります音声認識結果データから作成されます。

例)

```
TDMTCommand="10 (switch-if-level :MORPH)"
TDMTCommand="21 (switch-mode :J-E)"

UTTERANCE=1
NBEST=2
ORDER=1
wordids=5/10031/6
ORDER=2
wordids=5/10525/10002+10003/6
```

■ **TDMTCommand="10 (switch-if-level :MORPH)"**

翻訳方法の指定です。ATR-MATRIX では、形態素列を入力データとして翻訳をおこないます。MORPH は、形態素列を意味します。

■ **TDMTCommand="21 (switch-mode :J-E)"**

翻訳方向の指定です。ATR-MATRIX では、「日英」、「英日」、「日独」、「日韓」、「日中」の翻訳が可能です。J-E は日英方向翻訳（入力データが日本語で、英語に翻訳する）をあらわします。他の翻訳方向の指定は以下の通りです。どの方向に翻訳するかは、MATRIX カーネルがコンフィグレーションファイルから情報を取得し決定します。（MATRIX カーネルコンフィグレーションファイル仕様書参照）

英日翻訳	TDMTCommand="21 (switch-mode :E-J)"
日独翻訳	TDMTCommand="21 (switch-mode :J-G)"
日韓翻訳	TDMTCommand="21 (switch-mode :J-K)"
日中翻訳	TDMTCommand="21 (switch-mode :J-C)"

■ **UTTERANCE=N**

データのシリアル番号です。

■ **NBEST=N**

N が複数の場合複数の言語翻訳入力データが存在すると判断します。TDMT はこの全てを翻訳します。

■ **ORDER=N**

何番目の言語翻訳入力データかをあらわします。入力データが 1 つしかない場合、ORDER は 1 しかありません。例のように複数の言語翻訳入力データがある場合は 1 ~ N (NBEST の数) の間の値を取ります。

■ **wordids=5/...../6**

言語翻訳入力データです。データは形態素列であらわされています。この情報は音声認識結果データから取得します。(音声認識結果データ参照)

5 言語翻訳結果データ

言語翻訳結果データは言語翻訳モジュール TDMT の出力データです。この出力データには詳細な言語翻訳結果の情報が含まれています。このデータは加工され GUI 表示用データや、音声合成用入力データになります。

例)

```
TDMTCommand="10 (switch-if-level :MORPH)"
TDMTResult="10 :MORPH"
TDMTCommand="21 (switch-mode :J-E)"
TDMTResult="21 #<TM J-E>"
### TDMT_INPUT_FILTER: Language=J

UTTERANCE=1
NBEST=2
ORDER=1
wordids=5/10031/6
tdmt_pmorph=UTT-START

SOURCE_SENT="はい"
SYN_SENT="yes|INTERJ||"
SYN_SENT=:END
tdmt_nbest=1
tdmt_order=1
TARGET_SENT="Yes"
tdmt_score=0.0
tdmt_max_words=0
tdmt_min_words=1000
tdmt_partial=":PARTIAL|0|1|¥"はい¥"|¥"はい¥"|¥"Yes¥"|0.0|"
tdmt_partial=:END

tdmt_pmorph=UTT-END

ORDER=2
wordids=5/10525/10002+10003/6
tdmt_pmorph=UTT-START

SOURCE_SENT="ニューワシントンホテルでございます"
```

```

SYN_SENT="this|DET||"
SYN_SENT="is|BEV||"
SYN_SENT="the|DET||"
SYN_SENT="New Washington Hotel|PROPN||"
SYN_SENT=:END
tdmt_nbest=1
tdmt_order=1
TARGET_SENT="This is the New Washington Hotel"
tdmt_score=0.0
tdmt_max_words=0
tdmt_min_words=1000
tdmt_partial=":PARTIAL|0|2|¥"ニューワシントンホテル でございます¥"|¥"(ニューワシントンホテル ござい
ます)¥"|¥"This is the New Washington Hotel¥"|0.0|"
tdmt_partial=:END

tdmt_pmorph=UTT-END

comment=END-OF-UTTERANCE-1

.....キーワード

```

上記の言語翻訳結果データである TDMT の出力データの内、キーワードとなっている部分は最低限必要です。他の部分はなくともかまいません。では、そのキーワードの部分がどのように使われているかを説明します。

■ UTTERANCE=N

データのシリアル番号と同時に、必要なデータの始まりを意味します。このデータを使用するあらゆるコントローラーで必要となっています。

■ NBEST=N

出力結果の数です。TDMT コントローラーは SPREC の音声認識結果が複文の場合、一度に複数の文章を翻訳するように TDMT にまとめてデータを入力します。例の場合、「はい」と、「ニューワシントンホテルでございます。」の2文の複文ですので、出力言語翻訳結果の数は2つです。

■ **ORDER=N**

NBEST の項で説明いたしました通り、認識結果が複文の場合は複数の翻訳結果が出力されます。ORDER 数はこの複数の出力結果の何番目かをします。NBEST、ORDER ともに入力にいくつの音声認識結果データを入れたか？と言う事であって出力は関係ありません。この 2 つの文字列は、すべての翻訳データを抽出するためのキーワードとして利用されています。

■ **wordids=5/...../6**

TDMT 体系形態素列の音声認識結果であり、言語翻訳の入力データです。TDMT 体系形態素列表示に使用します。音声認識結果データから抽出したデータを使用しています。

■ **SYN_SENT="....." ~SYN_SENT= : END**

音声合成時に使用する GFILTER (言語フィルター) 用入力データです。品詞の種類や「読み」、アクセント情報が記述されています。

■ **tdmt_nbest=N**

前述 "wordids=" の入力に対し翻訳結果がいくつ存在するかを示します。すべての翻訳データを抽出するためのキーワードとして利用されています。最終的に翻訳結果の数は、【NBEST=N の N × tdmt_nbest=N の N】となります。

■ **tdmt_order=N**

1 つの入力音声認識データに複数の翻訳結果がある場合、N は 2 以上になります。これも、すべての翻訳データを抽出するためのキーワードとして利用されています。

■ **tdmt_pmorph=UTT-END**

1 つの言語翻訳結果の終わりを意味します。

■ **comment=END-OF-UTTERANCE-N**

すべての言語翻訳結果の出力が完了した事を意味します。つまり、データの最後です。

6 音声合成用データ

音声合成用データは音声合成モジュール CHATR への入力データです。CHATR への入力は言語翻訳結果データから作成されます。ただし、現在、英語以外の言語の文章は CHATR に対して直接入力する事ができません。そのため、多国語の音声合成用データを作成する言語フィルター (G FILTER) または、「かな漢字混じり文」を「ローマ字」に変換するソフトウェア (kan2rom) を使用しています。

上記の二つのプログラムは以下のような違いがあり、システム構成などによって自動的に切り替わりが行われています。

プログラム	対応言語	入力データ	使用方法	使用条件
G FILTER	日、独、韓、中国語	TDMT からの出力結果 (音声合成用データ)	MATRIX カーネル中の関数として呼び出す。	ATR-MATRIX 単体でのデモ、ATR-MATRIX 同士の接続時
kan2rom	日本語のみ	かな漢字混じり文	MATRIX カーネルが子プロセスとして実行	他翻訳システムとの接続時

言語フィルター (GFILTER) を使った音声合成用データの作成

TDMT から出力される音声合成用のデータは、前述の言語翻訳結果データの内、以下の部分です。

例)

```
SYN_SENT="#|3||"  
SYN_SENT="ニューワシントンホテル|普通名詞|にゅーわしんとんほてる  
|1100001000000080"  
SYN_SENT="です|助動詞|です|0"  
SYN_SENT="、|特殊||"  
SYN_SENT="どのような御用件でしょうか|感動詞|どのような ごようけん  
でしょうか|7100001000000011 7100001000000041"  
SYN_SENT="。|特殊||"  
SYN_SENT=:END
```

このデータを音声合成用データ作成フィルターである言語フィルター (GFILTER) に入力するフォーマットに変換します。この内容は言語毎に異なりますが、フォーマットは同じです。

例)

JM#|3||
ニューワシントンホテル|普通名詞|にゅーわしんとんほてる
|1100001000000080
です|助動詞|です|0
、|特殊||
どのような御用件でしょうか|感動詞|どのような ごようけんでしょうか
|7100001000000011 7100001000000041”
。|特殊||

言語情報 話者情報 表層 | 品詞 | 読み | アクセント情報 ¥ n
表層 | 品詞 | 読み | アクセント情報 ¥ n
:
表層 | 品詞 | 読み | アクセント情報 ¥ n

最初に言語情報と話者情報、以降が言語別翻訳結果の情報が格納されています。

- 言語情報 [J, E, G, K, C] J:日本語 E:英語 G:ドイツ語 K:韓国語 C:中国語
- 話者情報 [M, F] M:男性 F:女性

なお、言語ごとのフォーマットの詳細や、GFILTERの変換処理の詳細については、下村 賢 他著
“TDMTとCHATRのインターフェース：言語フィルタ” TR-IT-0354)を参照してください。

GFILTERからは以下のようなデータが出力されます。

例)

nyuuwashiNtoNho'terudesu l do'noyounagoyoukeNdeshouka 5

kan2rom を使った音声合成用データの作成

日本語の場合には GFILTER を使わずに「かな漢字文字列」を直接「ローマ字」に変換する事ができます。これは、TDMT の言語翻訳出力結果を使わない音声合成時に有効です。この時変換ツールとして「kan2rom」を使用します。

何人で旅行されますか？



na'NniNde 2 ryokoo 1 saremasUka 5

CHATR 入力用音声合成データのフォーマット

このようにして得られた音声合成用データを CHATR に入力できる形に変換します日本語、ドイツ語、韓国語、中国語の場合はフォーマットが異なります。

日本語 : (Say (Synth (Utterance Phoneme (.....))))
アクセント処理、音素表記変換は mor2bou でおこないます。

ドイツ語 : (Say (Synth (Utterance PhonoForm (.....))))
アクセント処理、音素表記変換は GFILTER でおこないます。

韓国語 : (Say (Synth (Utterance Phoneme (.....))))
アクセント処理、音素表記変換は HCODE でおこないます。

中国語 : (Say (Synth (Utterance Phoneme (.....))))

※.....は、GFILTER、kan2rom で得られた音声合成用データです。

7 音声認識結果表示用データ

音声認識結果表示用データは音声認識結果データから抽出、作成されます。ですから、データの形式は音声認識結果データと何ら変わりません。文字列を GUI で表示するときには、音声認識結果データをそのまま GUI コントローラーへ送信します。

GUI コントローラーは音声認識結果データから、以下の部分の行を抽出します。

例)

```
WORDS=UTT-START/SENT-START/はい/SENT-END/SENT-START/ニューワシントンホテル/で+  
ございます/SENT-END/UTT-END
```

GUI コントローラーは GUI に対して表示するようにコマンド化して送信します。

例)

```
japan.textwindow draw -text はい ニューワシントンホテルでございます。  
-foreground blue ¥n
```

8 言語翻訳結果表示用データ

言語翻訳結果表示用データは言語翻訳結果データから抽出、作成されます。ですから、データの内容は言語翻訳結果データと何ら変わりません。文字列を GUI で表示するときには、言語翻訳結果データをそのまま GUI コントローラーへ送信します。

GUI コントローラーは言語翻訳結果データから、以下の部分の行を抽出します。

例)

```
TARGET_SENT="Yes"  
TARGET_SENT="This is the New Washington Hotel"
```

GUI コントローラーは GUI に対して表示するようにコマンド化して送信します。

例)

```
us.textwindow draw -text Yes This is the New Washington Hotel -foreground  
red ¥n
```

9 対コミュニケーションサーバー通信データ

対コミュニケーションサーバー通信データは、CMU（カーネギーメロン大学）が開発した、CSTAR-II 実験用通信サーバーシステムです。このコミュニケーションサーバーと翻訳結果のやり取りをおこないます。

■ CSへの送信データフォーマット

Message 種類△言語種類△UTTERANCE番号△{ データ }

■ CSからの受信データフォーマット

Message 種類△言語種類△UTTERANCE番号△{ データ }-src 送信元サイトID

△ 半角スペース

10 GUIコマンド

GUI コマンドとは、SipGuiController と GUI 間でのコマンドのやりとりを、行うための文字列です。GUI の制御や、ユーザーからの ATR-MATRIX カーネルステータス変更指示などのコマンドがあります。

詳細は、松井 孝典 他著「ATR-MATRIX GUI 仕様書」(TR-IT-0347) を参照して下さい。

索引

A

accept	6
amname	7, 8
awake	6

B

button	6
byebye	6

C

CHATR	15, 17
check	6
connect	6
C S	20
csTrans	6

D

divs	7, 9
------	------

E

end	6
error	6
exit	6

G

G FILTER	15
GFILTER	14, 16, 17
GUI	4, 7, 8, 12, 18, 19, 21
GUI コマンド	4, 21
GUI 仕様書	21

H

HCODE	17
hello	6
host	6

I

initialize	6
------------	---

K

kan2rom	15, 17
---------	--------

L

language	6, 7
log	6

M

MORPH	10, 12
-------	--------

N

NBEST	7, 8, 10, 11, 12, 13, 14
-------	--------------------------

O

ORDER	7, 8, 10, 11, 12, 14
-------	----------------------

R

ready	6
recog	6
result	6

S

SipChatrController	5
--------------------	---

SipConnectController	5
SipGuiController	5, 21
SipMainController	5
SipSoundController	5
SipSprecController	5
SipTdmController	5
sleep	6
speaker	6
SPREC	7, 8, 13
start	6
SYN_SENT	12, 13, 14, 15

T

TDMT	8, 10, 12, 13, 14, 15, 16, 17
tdmt_nbest	12, 13, 14
tdmt_order	12, 13, 14
tdmt_pmorph	12, 13, 14
trans	6

U

UTTERANCE	7, 8, 10, 12, 13, 14
-----------	----------------------

W

wordids	7, 8, 10, 11, 12, 14
WORDS	7, 8, 18

あ

アクセント情報	14, 16
---------	--------

え

英語	16
----	----

お

音声合成	3
音声合成用データ	4, 15, 16

音声認識	3
音声認識結果	4, 7, 8, 10, 11, 13, 14, 18
音声認識結果データ	4, 7, 8, 14, 18
音声認識結果表示用データ	4, 18
音素表記変換	17

か

カーネギーメロン大学	20
かな漢字混じり文	15
かな漢字混じり文字列	8
かな漢字文字列	17
韓国語	3, 16, 17

け

言語フィルター	14, 15, 16
言語翻訳	3
言語翻訳結果データ	4, 12, 13, 15, 19
言語翻訳結果表示用データ	4, 19
言語翻訳入力データ	7, 10, 11

こ

コミュニケーションサーバー	4, 20
コントローラー	4, 5, 6, 8, 13, 18, 19
コントローラー間通信パケットデータ	4, 5

し

女性話者	8
------	---

た

対コミュニケーションサーバー通信データ	4, 20
男性話者	8

ち

中国語	3, 15, 16, 17
-----	---------------

つ

通信パッケージ 5

と

ドイツ語 3, 16, 17

に

日本語 3, 16, 17

は

発音記号データ 9

ほ

翻訳方向 10

ゆ

尤度 8

ろ

ローマ字 15, 17

わ

話者選択 8