

TR-IT-0299

CHATR の Perl による拡張 An Extension of CHATR Using Perl

芦村 和幸
Kazuyuki Ashimura

1999.3

音声合成システム CHATR を利用する際、いままで、いくつかの不便な点があった。筆者は、CHATR のモジュール整理について検討してきたが、CHATR は基本的に研究用ワークベンチであるため、「改造のベースライン」を特定するのが困難である。また、今現在安定して動作している CHATR 自体を改造することは、デグレード発生の危険をもはらんでいる。そこで、CHATR 自体は改造せずに、「CHATR に含まれるさまざまな処理を、あたかも API のように利用するための拡張モジュール」を Perl により作成した。さらに、上記の拡張モジュールを利用して、CGI による TTS システムを作成するとともに、かねてより課題であった、合成音における音韻継続時間長設定の改善について検討した。

CHATRのPerlによる拡張

TR-IT-0299

目次

| | | |
|-----|-------------------------|----|
| 第1章 | 多言語音声合成システム CHATR | 2 |
| 第2章 | CHATR 利用時の問題と対処..... | 3 |
| 第1節 | CHATR を利用する際の問題点..... | 3 |
| 第2節 | 問題への対処方針..... | 4 |
| 第3節 | 対処の手順..... | 4 |
| 第3章 | PERL による拡張の処理構成 | 5 |
| 第1節 | 基本的な考え方..... | 5 |
| 第2節 | 具体的な処理構成..... | 5 |
| 第4章 | 拡張1：CHATR-CGI | 8 |
| 第1節 | ポイント | 8 |
| 第2節 | インストール方法..... | 9 |
| 第3節 | 利用方法 | 11 |
| 第5章 | 拡張2：音韻継続時間長改善..... | 12 |
| 第1節 | ポイント | 12 |
| 第2節 | 実装にあたっての問題と対処..... | 13 |
| 第3節 | CHATR のチューン..... | 14 |
| 第4節 | 効果確認 | 15 |
| 第5節 | インストール方法..... | 17 |
| 第6節 | 利用方法 | 17 |
| 第6章 | 謝辞 | 18 |

[オンライン情報について]

本ドキュメントのオンライン版が以下のURLにあります。補足情報として、使用したスクリプト類のソースリストも記述してありますので、ぜひ、ご参照ください。

<http://itlpc60/~kasimura/chatr/TR/index.html>

第1章 多言語音声合成システム CHATR

ATR 音声翻訳通信研究所では、ある話し手の声を一定量以上（例えば 1 時間程度）、コンピュータに記憶して音声データベース化しておき、その中に多数ある同じ種類の音韻（子音や母音）の中から、出力したい文に応じて、音韻の並び方や声の高さ、音韻の長さなどの条件が良く適合し、しかも滑らかにつながる音韻を選び出し、それらの音声波形をつなぎ合わせるという方式の音声合成システム CHATR を開発しました。

これにより、短期間で「その人らしい声」を合成することが可能となりました。この方式は基本的に言語や話し手に依存しないので、あらゆる言語に適用が可能であり、現在のプロトタイプでは、日本語・英語・韓国語・ドイツ語・中国語の 5 カ国語が合成できると同時に、日本語については黒柳徹子さんの声をはじめ数十名の声を使った合成音声を出力することが可能です。

第2章 CHATR 利用時の問題と対処

第1節 CHATR を利用する際の問題点

- CHATR は、「音声第二研で音声合成の研究を行うためのワークベンチ」として利用するために、「単なる音声合成器」としてではなく、いろいろなモジュールを切替えて試することができる、一種の開発環境として実装されています。
- CHATR では、「Lisp ライクなコマンド群 (以下、CHATR コマンド)」により、CHATR ワークベンチ中に実装されている様々なモジュールの組合せを変えたり、動作の微調整をするために各種パラメータ値を変更することができます。しかし、CHATR コマンドは、複雑な音声合成処理の組合せを制御するために、コマンド体系としてかなり複雑になっています。そのために、CHATR を利用する場合、以下に示すように、いくつかの不便な点がありました。

| 利用の目的 | 不便な点 |
|----------------|--|
| 音声合成器として応用 | <ul style="list-style-type: none">● 単純に定型的な音声合成を行いたい場合でも、複雑な CHATR コマンドを習得する必要がある。● 各種モジュールの呼び出し方が API としては整理されていないため、システムへ組み込む際、単純な関数コールではモジュールを呼び出せない。● CHATR コマンドにも、本格的なプログラミング機能があるが、実際に複雑な処理を行う場合は、使い慣れた Shell スクリプトなどを使って処理を制御した方が便利ことが多い。しかし、CHATR の外部から Shell スクリプトなどを利用して毎回 CHATR を起動させると、処理速度が非常に遅くなる。 |
| 研究用ワークベンチとして利用 | <ul style="list-style-type: none">● ソースコードは C 言語で記述してあるが、約 50000 行と規模が大きいため、扱いにくい。● 実際に、研究やチューニングのために、C 言語のソースコードを書き換える際、「Lisp インタプリタの機能と音声合成機能との切り分けが不明確」など、処理の切り分けや構成に問題があるため、「具体的には、どこに何を追加すればよいか」がわかりにくい。 |

第2節 問題への対処方針

- CHATR を利用していく際の問題を解決するためには、CHATR を「使いやすく、わかりやすく、チューンしやすく」していく必要があります。
- しかし、CHATR は基本的に研究用ワークベンチであり、日々、進化しているため、「改造のベースライン」を特定するのが困難です。また、今現在安定して動作している CHATR 自体を改造すると、デグレードが発生する心配があります。
- そこで、CHATR 自体を改造するのではなく、「今ある CHATR を、うまく活用する方法」を考えるべく、以下のような対処方針を設定しました。
 1. できるだけ CHATR の外部から、音声合成処理を制御する。
 2. 処理、制御、チューニングの3つを完全に切り分けることにより、使いやすく、わかりやすく、チューンしやすいものにする。
 3. 今後のライブラリ化、API 化の指針となるように、処理構成を構造化する。

第3節 対処の手順

- 具体的には、以下の3ステップにより、上記の対処を行いました。

| ステップ | ねらい | 作業内容 |
|------|----------------|---|
| 1 | とにかく、使えるものを作る。 | ● オープンハウスのデモのために、Perl を用いた CGI により、「誰でも簡単に使える TTS システム」を作成する。 |
| 2 | わかりやすく構造化する。 | ● チューニングや拡張をやりやすいように、CGI スクリプトの処理構成を再検討し、クラス化やモジュール化を行う。 |
| 3 | 実際にチューニングを加える。 | ● 処理、制御、チューン箇所が切り分けられ、構造化されたモジュールを利用して、「CHATR の韻律 (F0 制御や音韻継続時間長制御)」のチューニングを行う。 |

第3章 Perl による拡張の処理構成

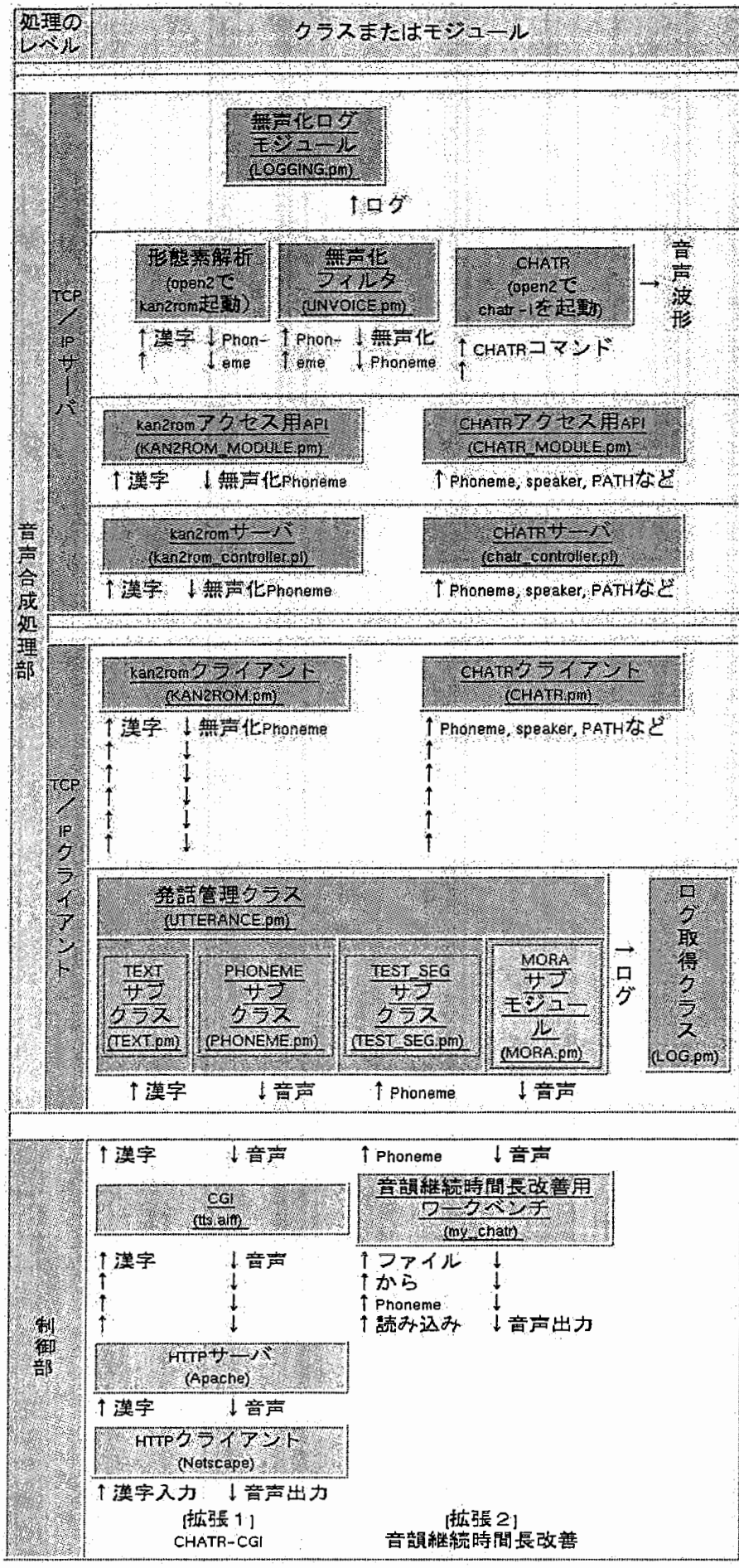
第1節 基本的な考え方

- Perl により、以下の2種類の拡張を行いました。
 - 拡張1： CHATR-CGI
 - 拡張2： 音韻継続時間改善用のワークベンチ
- 上記2種類の拡張にあたって、Perl Ver.5 のクラス化機能を利用して、「制御部」と「音声合成処理部」を完全に切り分けたモジュールを作成したため、2種類の拡張は、「制御部」の構造のみが異なり、「音声合成処理部」は全く同一のものを共用することが可能になりました。
- さらに、「音声合成処理部」を、以下の2つに切り分けました。
 1. 形態素解析、母音の無声化、音声合成などの具体的な処理を行う TCP/IP サーバ
 2. 音声合成の対象になる発話内容を管理する TCP/IP クライアント
- これにより、音声合成に必要な研究的な処理は、全て「TCP/IP サーバ」が独自に行うことになるため、利用者がこれらの研究ソフトの処理内容を関知する必要は、一切なくなりました。

第2節 具体的な処理構成

- 具体的な処理構成について、次ページの図に示します。

処理構成図



※注意：

TTS システム全体のターンアラウンドを向上させるために、音声波形は、以下の流れで扱うようにしました。

1. 「TCP/IP サーバのハードディスク」を、あらかじめ、TCP/IP クライアントへ export しておく。
2. 「TCP/IP サーバ上の CHATR」で作成した音声ファイルは、「TCP/IP クライアントへ export されたハードディスク」に書き込む。
3. 音声出力にあたっては、「TCP/IP クライアント」が、「TCP/IP クライアントへ export されたハードディスク」から自動的に音声波形を読み出した上で、音声を再生する。

第4章 拡張 1 : CHATR-CGI

第1節 ポイント

- CHATR-CGI は、オープンハウスのデモとして、「誰にでも使いやすい音声合成アプリケーション」を目指して作成しました。
- Web 経由で、日/英の2種類の TTS を行うことができます。日本語については、漢字入力と Phoneme 入力に対応しており、英語については、TEXT 入力に対応しています。
- 合成したい文章を入力し、ボタンを押すだけで、音声合成が行えます。Netscape など、一般的なしくみを流用しているのも、誰にでも直観的に使いやすく、また、ブラウザさえあればどこからでも利用できます。
- 「制御部、発話管理、TCP/IP クライアント」を複数用意すれば、「一つの TCP/IP サーバ部」を使って、複数のネットワークに対して、音声合成サービスを提供できます。例えば、同じ一つのサーバを使って、「オープンハウスなど ITL 内部だけのデモ」も「WWW 経由での、全世界に対するデモ」も提供できます。
- 「母音の無声化」や「単位選択パラメータの調整」などのチューニングを、サーバ側で集中的に管理できるメリットもあります。

第2節 インストール方法

1. 各スクリプトは、CGI として利用できるディレクトリに インストールする必要があります。

itlpc60 と atra12(www.itl.atr.co.jp)のインストール内容

| マシン | 制御部 発話管理 TCP/IP クライアント | TCP/IP サーバ |
|---------|---|---|
| Itlpc60 | /home/httpd/cgi-bin/kasimura/interface/ | /home/httpd/cgi-bin/kasimura/interactive/ |
| Atra12 | /home/atra07/chatr/cgi/cgi-oshaberi/ | — |

※注意：

atra12 のクライアント部も、 itlpc60 のサーバ部を利用して動作するため、 atra12 には「TCP/IP サーバ」はインストールされていません。

itlpc60:/home/httpd/cgi-bin/kasimura/interface/

および

atra12:/home/atra07/chatr/cgi/cgi-oshaberi/

の内容

| ファイル名 | 内容 |
|--------------|---|
| CGI.pm | CGI を利用するためのクラス http://waldo.wi.mit.edu/WWW/tools/scripting/CGI_libraries/CGI.pm/ より取得 |
| CHATR.pm | CHATR 利用のための TCP/IP クライアント |
| KAN2ROM.pm | kan2rom 利用のための TCP/IP クライアント |
| LOG.pm | 発話管理クラス(UTTERANCE.pm)のログ取得を行うクラス |
| MORA.pm | 発話管理クラス(UTTERANCE.pm)のために、 モーラ単位の処理を行うサブモジュール |
| PHONEME.pm | 発話管理クラス(UTTERANCE.pm)の代わりに、 Phoneme レベルの処理を行うサブクラス |
| TCPIP.pm | TCP/IP 通信のサーバとクライアントを接続するためのライブラリモジュール |
| TEST_SEG.pm | 発話管理クラス(UTTERANCE.pm)の代わりに、 test_seg レベルの処理を行うサブクラス |
| TEXT.pm | 発話管理クラス(UTTERANCE.pm)の代わりに、 テキストレベルの処理を行うサブクラス |
| UTTERANCE.pm | CHATR 利用にあたって、合成対象である発話を管理 するクラス |
| english_in | 英語 TTS のテストデータ |
| japanese_in | Phoneme 入力の日本語 TTS のテストデータ |
| kanji_in | 漢字入力の日本語 TTS のテストデータ |
| tts.aiff | CGI スクリプト本体 |

itlpc60:/home/httpd/cgi-bin/kasimura/interactive/
の内容

| ファイル名 | 内容 |
|-----------------------|---|
| CHATR_MODULE.pm | open2 で起動した CHATR にアクセスするための API ライブラリ |
| KAN2ROM_MODULE.pm | open2 で起動した kan2rom にアクセスするための API ライブラリ |
| LOG.pm | CHATR サーバ(chatr_controller.pl)と kan2rom サーバ(kan2rom_controller.pl)のログ取得を行うクラス |
| LOGGING.pm | 無声化フィルタ (UNVOICE.pm)のログ取得を行うサブモジュール |
| TCPIP.pm | TCP/IP 通信のサーバとクライアントを接続するためのライブラリモジュール |
| UNVOICE.pm | 母音の無声化フィルタ |
| chatr_controller.pl | CHATR サーバ |
| kan2rom_controller.pl | kan2rom サーバ |

2. 次に、CGI スクリプトを実行するために、Perl のパス名を確認し、必要であれば、「tts.aiff」の1行目のパス名指定を変更します。以下に、itlpc60 と atra12(www.itl.atr.co.jp)の Perl のパス名を示します。

| | |
|---------|--|
| Itlpc60 | /usr/bin/perl |
| Atra12 | /usr/local/lang/perl-5.004_04/bin/perl |

3. 最後に、CGI を動作させる前に、itlpc60:/export/itlpc60_data/oshaberi/rm.sh を起動して、不要なログファイルを消去しておいてください。
CGI は Nobody 権限で実行されるため、もし、ログファイルが残っていると、書き込みエラーが発生する場合があります。

第3節 利用方法

1. CGI の利用方法は簡単です。まず、CHATR サーバおよび kan2rom サーバを起動してください。

| | |
|-------------|--|
| CHATR サーバ | <code>\$ cd /home/kasimura/cgi-bin/interactive; ./chatr_controller.pl</code> |
| kan2rom サーバ | <code>\$ cd /home/kasimura/cgi-bin/interactive; ./kan2rom_controller.pl</code> |

2. 次に、Netscape などのブラウザから、以下の URL へアクセスしてみてください。

`http://www.itl.atr.co.jp/chatr/interactive/index.html`

`http://itlpc60/~kasimura/chatr/interactive/japanese/index.html`

CGI の場合は、必要なスクリプト類が、HTTP サーバから自動的に起動されます。うまく CGI が使えない場合は、TCP/IP サーバが起動されていない可能性があります。もしサーバが起動していてもうまく動かない場合は、TSG に相談してください。

3. なお、HTTP サーバを経由せずに、デバッグモードで CGI を動作させるには、CGI がインストールされたディレクトリで以下のようにしてください。

```
$ ./tts.aiff <japanese_in
```

第5章 拡張2：音韻継続時間長改善

第1節 ポイント

- 以前から、ポーズ長の設定については、ポーズのラベリングを3分割することにより、改善がみられていました。
ポーズ改善については、以下の URL を参照してください。
<http://itlpc60/~kasimura/chatr/p1p2p3/0807.CHATR-ALPHA.p1p2p3/>
- 一方、ポーズ以外の通常の音素については、以前、「単位選択基準に、モーラ単位での時間長設定誤差を追加」しましたが、あまり効果がみられませんでした。
単位選択基準への、モーラ長の組み込みについては、以下の URL を参照してください。
<http://itlpc60/~kasimura/chatr/duration/index.shtml>
- CHATR は、「音素環境 (スペクトルに相当)」や「音の高さ、長さ」を基準として、音声 DB 中から、韻律予測処理で求めた韻律ターゲットに最も近い音素を選択して合成音を生成します。音声 DB 中の音素は有限なので、「音素環境」、「音の高さ」、「音の長さ」は、どれかを優先度を上げると、他の優先度が下がるという、二律排反の関係です。
- したがって、「音韻継続時間長を単位選択基準として優先すること」は、「音の高さ」や「波形の接続」へ悪影響を与える可能性があります。
- そこで、「音韻継続時間長を単位選択基準として優先すること」は、いったんあきらめ、単位選択基準以外の部分で補うために、以下の2通りのチューニングを行うことにしました。

| | |
|-------|---|
| チューン1 | <ul style="list-style-type: none">● CHATR の単位選択では、Viterbi アルゴリズムにより、最もコストの小さい音素接続を探しているが、候補数やビーム数の設定を大きくすると、長い文の場合、計算量が爆発する。● 一方、逆に候補数が不十分な場合などは、「とび抜けてコストの高い候補」が混ざっていても、全体コストとしては、さほど大きくなり、コストの高い候補を選択してしまう可能性もある。そのため、単位選択を、呼気段落ごと (=ポーズ区切りごと) に分割して行ってみる。 |
| チューン2 | <ul style="list-style-type: none">● 音韻継続時間長については、単位選択よりももっと遅い時点 (=波形接続の後処理の時点) で、時間長を再評価し、ターゲットよりも長い場合は、母音やポーズなどのスペクトルの安定している箇所サンプルを間引き、ターゲットに厳密に合わせる。 |

第2節 実装にあたっての問題と対処

| 項番 | 問題 | 対処 |
|----|---|--|
| 1 | <ul style="list-style-type: none"> 「チューン1」のために、韻律予測結果を、Segments 形式の中間ファイルで Save した上で、呼気段落ごとに分離している。 しかし、韻律予測結果は話者 MHT のモデルから計算した情報であり、また、通常の Segments 形式では、計算結果の韻律情報がそのまま出力されるため、話者 MHT の韻律情報が Save されてしまう。 このため、MAK など他の話者で合成音を作成すると、ターゲット韻律が正しくないことになってしまう。 | <ul style="list-style-type: none"> 韻律予測結果から話者依存性を除くために、生データでなく、Zscore で Save する Segments 形式 (Z-seg) を作成する機能を CHATR に追加した (chatr-0.95 および chatr-alpha.kasimura に実装済み)。 |
| 2 | <ul style="list-style-type: none"> 「チューン2」で、波形の間引きを行う際、「ターゲットより長いかどうか」を生データを用いて判定し、長い場合は、サンプル単位で間引いている。 この方式では、「ターゲットより短い音はそのまま」であり、「ターゲットより長い音はターゲット通りに切る詰める」ため、全体としては、ターゲットより短くなる (=早口になる) 傾向にあると思われる。 「早口になる」といった問題は特に発生していないが、「モーラ単位の生データ値」を用いた時間長判定では、「聴覚上間延びして聴こえる箇所」を洗い出すことができず、波形切り詰め効果も得られていない。 | <ul style="list-style-type: none"> 「音素長をモーラごとに積算」した上で、Zscore を求め、「モーラ単位の Zscore」により、「ターゲットより長いかどうか」を判定するようにしたところ、「聴覚上間延びして聴こえる箇所」である、「不必要に長い長母音」を洗い出すことができるようになった。 「モーラ単位の時間長の Zscore」による判定の効果については、「第4節 効果確認」を参照のこと。 上記の作り込みにあたっては、波形の間引きをやりやすいように、あらかじめ、「ターゲットより長い候補」を優先して選ぶように、「CHATR の単位選択」にもチューンを加えた。 CHATR のチューン箇所については、「第3節 CHATR のチューン」を参照のこと。 |

第3節 CHATR のチューン

- 「チューン2」を行うにあたって、CHATR の以下のモジュールにチューニングを加えました。

| モジュール名 | チューニング内容 | 種別 |
|--------------------------|---|---------|
| udb_mora_len.{c,h} | モーラ単位で音韻継続時間長を再評価する。 | モジュール追加 |
| udb_check_neighbor.{c,h} | 音声 DB 中で実際に連続している音素接続については、接続コストを0にする。 | モジュール追加 |
| udb_nus_score.c | 「音韻継続時間長優先 (CHATR-0.95 と同じ)」と「接続性優先 (CHATR-ALPHA と同じ)」という、2種類の音声単位切り出し位置シフトの方法を、切替えて使えるように変更。 | モジュール変更 |
| udb_generic.c | モジュール追加への対応、パラメータ取得の追加 | モジュール変更 |

- 上記のチューニングを加えた CHATR のソースコードは、

`itlpc60:/export/itlpc60_data/CHATR/chatr-alpha.kasimura/src/udb`

にあります。

第4節 効果確認

- MAKp の DB を使って実際に合成音を作成し、「ポーズ区切りごとの単位選択」と「モーラ単位の時間長の Zscore に基づく波形の間引き」による、音韻継続時間長設定改善の効果を確認しました。

第1項 入力情報

| | |
|----------|---|
| 漢字かな混じり文 | 9日未明大分・佐賀関町のJR日豊線の幸崎駅で 停車中の電車の屋根で遊んでいた高校生ら少年・少女5人が 感電し大やけど、うち1人が重体。 |
| Phoneme | kokonokamimee 3 oo'ita 3 saganoseki'choono 3 jee'a'aru 2 nipooseNno 2 koosakie'kide 4 teeshachuuno 3 deNshano 2 ya'nede 2 asoNdeita 3 kooko'oseera 3 shooneNsho'ojo 2 goni'Nga 3 kaNdeNshi 3 ooya'kedo 4 uchi 1 hIto'riga 2 juutai 5 |

第2項 合成結果

| 作成条件 | 合成結果 |
|--|--|
| [基準] 通常の CHATR | <ul style="list-style-type: none"> ● 「遊んで」が、/aso'Nde/ ● 「停車」の/ee/が長過ぎる。 ● 波形の接続は良好。 |
| [チューン1] 単位選択を、 ポーズ区切りごとに分割して行う | <ul style="list-style-type: none"> ● 「遊んで」が、/aso'Nde/ ● 「高校生ら」が、/kookoose'era/ ● 「停車」の/ee/が長過ぎる。 ● 波形の接続は少し悪い箇所がある。 |
| [チューン2] チューン1 + 「モーラ単位の時間長の Zscore」 にもとづいて波形の間引く | <ul style="list-style-type: none"> ● 「遊んで」が、/aso'Nde/ ● 「高校生ら」が、/kookoose'era/ ● 「停車」の/ee/が少し長い。 聞いた印象としては、まだ少し長いですが、音韻継続時間長自体は、「間引き」の効果により 20ms 短くなっている。 ● 波形の接続は少し悪い箇所がある。 |

第3項 まとめ

- CHATR の単位選択結果である UnitLabels2 情報から「モーラ単位の時間長の Zscore」を計算し、これにもとづいて音韻継続時間長の妥当性を判定すると、以下のような音素において、時間長が長すぎる箇所を洗い出すことができました。

| 音素の種別 | 間引かれた音素の数 |
|---------------------------|-----------|
| 長音 | 11 |
| 撥音 | 5 |
| 独立した母音 {a,i,I,u,U,e,o} | 4 |

- しかし、波形の間引きを行っても、聴覚上は、まだ「間延びして聞こえる箇所」が残っていました。これについては、「モーラ単位の時間長の Zscore」により時間長の妥当性を判定を行う際のしきい値を調整することにより改善していけるのではないかと考えられます（現在は、「Zscore で 0.1 以上の誤差があるとき」に、許容範囲外としています）。
- また、このチューニングにより波形の接続が少し悪くなったような印象があるため、今後、接続も考慮した改善方法を検討していく必要があります。さらに、このチューニングは、時間長設定に特化しているため、アクセントやイントネーションの改善についても、別途、検討する必要があります。

第5節 インストール方法

- 音韻継続時間長改善ワークベンチは、「音声合成処理部」としては、CHATR-CGI と全く同じものを利用しています。
- したがって、すでに CHATR-CGI がインストールされていれば、インストール作業は不要です。
- もし、CHATR-CGI をまだインストールしていない場合は、「第4章 拡張1」の「第2節 インストール方法」をご覧ください。

第6節 利用方法

- 音韻時間長設定改善ワークベンチの利用方法は、CGI の利用方法に少し似ています。まず、CHATR サーバおよび kan2rom サーバを起動してください。

| | |
|-------------|---|
| CHATR サーバ | \$ cd /home/kasimura/cgi-bin/interactive; ./chatr_controller.pl |
| kan2rom サーバ | \$ cd /home/kasimura/cgi-bin/interactive; ./kan2rom_controller.pl |

- 次に、スクリプト my_chatr を起動してください。

例えば、itlpc60 なら、以下のように起動します。

```
$ cd /home/kasimura/public_html/chatr/duration/cut_wave  
$ ./my_chatr kokonoka.phoneme MAKp
```

my_chatr の第一引数は、「合成対象 phoneme を含むファイル」で、第二引数は、「話者 DB 名」です。

※注意：

my_chatr では、発話内容をポーズごとに分割するため、話者 DB 名としては、「p」のついた DB を指定する必要があります。

第6章 謝辞

私の拙い研究活動を支えてくださった、ATR 音声翻訳通信研究所のみなさんに感謝いたします。

特に、以下の方々に深謝いたします。

- 研究の機会を与えてくださった、ATR 音声翻訳通信研究所の山崎泰弘前社長（現国際電信電話株式会社研究所）および山本誠一社長、そして第二研究室の樋口宜男前室長（現国際電信電話株式会社研究所）
- 研究を進めるにあたって、多大なご指導をいただきました、第二研究室のニック・キャンベル室長、および第一研究室の匂坂芳典室長
- CHATR を含めて、研究活動のあらゆる側面においてご支援をいただいた、第二研究室のみなさん