

作業効率の高いテキスト合成音声チューニング環境

立花 隆輝 長野 徹 西村 雅史

Text-to-Speech Synthesis System with an Effective and Easy-to-Use Tuning Interface

Ryuki Tachibana, Tohru Nagano, and Masafumi Nishimura

テキスト音声合成は任意のテキストを音声に変換する技術である。決まったテキストを音声に変換するオフライン・アプリケーションでは、手作業でチューニングを行うことで、自動生成した場合よりも場面によく適合し、自然性の高い音声を作成することができる。しかし、従来チューニング作業は専門知識が必要で手間のかかる作業であった。筆者らはグラフィカルなインターフェースを備えたクライアント・ソフトウェアと、複数ユーザーが同時接続可能なサーバー・ソフトウェアを開発することで、誰にでもチューニング作業の容易なシステムを構築している。このシステムの有効性を2つの大規模利用例と2つの実験結果によって示す。このシステムにより、望み通りの合成音声を作りやすくなり、e-Learning教材やpodcastなど、音声を含むコンテンツを誰でも手軽に作って発信できるようになる。

Text-to-speech (TTS) synthesis is a computer technology used to produce synthetic voices to read aloud input text. With offline applications, manual tuning can be used to create a more natural and appropriate synthetic voice. However, manual tuning has always been time-consuming endeavor that requires expertise in the field of TTS. Given this, we are developing a system that is composed of client software with a graphical user interface and server software to which multiple clients can simultaneously connect. In this paper, we describe the system design and demonstrate the system's effectiveness by looking at two case studies of it being used on a large scale and two sets of experimental results. With this system, non-expert users can easily generate and tune synthetic voices, as well as publish multimedia content such as e-learning materials and podcast series that make use of synthetic voices.

Key Words & Phrases : テキスト音声合成, チューニング, グラフィカル・ユーザー・インターフェース
Text-to-speech synthesis, tuning, graphical user interface

1. はじめに

テキスト音声合成は任意のテキストを音声に変換する技術である。合成音声にはいわゆるロボットのようなアクセントや声質のイメージを持っている人も多いことと思われるが、近年の大きな技術進歩によって精度や音質が向上し、アクセントや声質の点でも人間の肉声に近いものとなってきた。しかし、それでもまだ漢字の読みの推定や標準語アクセントの再現において少数の誤りが生じることは避けられない。また、人間のように文の意味内容に応じて感情表現豊かなイントネーションを付けることはできない。より豊富な計算機資源を使えばある程度の改善は可能だが、デスクトップPCで速やかに合成音声を生成することは難しい。

音声合成にはホームページ読み上げのようにリアルタイムで自動的に合成音声を生成することが求められるリアルタイム・アプリケーションもあるが、e-Learning教材やpodcastコンテンツのようにオフラインで合成音声を生成しておくことが許されるオフライン・アプリケーションも存在する。後者においては、合成音声を自動的に生成した後に、人手によって読みやアクセントの誤りを修正し、文の意味内容に応じたイントネーションを与えるというチューニング作業を行う余地がある。しかし、これまでその作業にはアルゴリズムやデータ構造などに関する専門知識が必要であり、また専門家にとってもツールが整備されていないため時間と手間がかかっていた。このような理由から、正しい読み・アクセントおよび望むようなイントネーションを持った合成音声の作成は、誰にでも速やかに、かつ簡単にできる作業ではなかった。

本論文では、この問題を解決するために次のような特徴を持ったシステムを提案する。

提出日:2009年9月7日 再提出日:2010年6月4日

- (1) 音声合成機能をクライアントとサーバーに分離し負荷を分散する。これによりサーバーでは豊富な計算機資源を使用する一方で、クライアントでは複数文に関するチューニング作業を並列して行うことができる。
- (2) クライアントには、テキスト解析結果や音声素片の情報をグラフィカルに表示し、インタラクティブに編集できるインターフェースを Adobe Flex 3 [1] によって構築する。
- (3) システムの各部品は、独立しても動作する再利用可能なコンポーネントとして設計し、その間の通信プロトコルを XML に基づいて定義する。これにより個々のコンポーネントを別の目的に再利用することも可能となる。

2. 合成音声チューニングの関連動向

2.1 IBM 社内での関連動向

IBM の音声合成アプリケーションとしては IBM ホームページ・リーダー [2] が社内外によく知られている。ここに応用されている音声合成技術は、デスクトップ音声合成ソフトウェア ProTalker 97 [3] に使われたのと同じ技術であり、計算機資源の点で効率化されているため、ホームページ・リーダーのようにダウンロード利用するアプリケーションにも適している。一方で、音質とアクセントの自然さにおいては改善の余地が残されていた。

現在 IBM で研究されている最新の音声合成技術はその手法とは異なり、より大規模の音声コーパス^{*1}を機械学習に用いることで優れた音質と自然なアクセントを実現している [4]。この技術は、IBM ソフトウェア・グループ (SWG) の WebSphere[®] Voice Server と Embedded ViaVoice に利用されており、開発時に想定した分野のテキストを合成する際、特に人間の肉声に迫る品質が実現できる。その一方で、想定分野外のテキストを合成する際、想定内テキストに比べて誤りや不自然性が多く見つかることがある。しかし、そのような場合にもオフライン・アプリケーションではチューニングで品質を改善できる。

筆者らは実際に、お客様がオフライン・アプリケーションで利用するテキストや、社内向け podcast コンテンツについてチューニングを施すことがある。その場合には従来、IBM eSPR (enhanced Symbolic Phonetic Representation) と呼ばれる複雑なデータ・フォーマッ

トを編集の容易な独自フォーマットに変換した上で、テキスト・エディターを用いて編集していた。この作業にはツールも十分整備されておらず専門知識が要求されていた。ソフトウェア・グループでもチューニング作業の必要性は認識しており、TTS Tuner [7] と呼ばれるソフトウェアを提供しているが、グラフィカルには編集を行うことができないので、一般ユーザーに使いやすいものではない。またテキスト解析結果のチューニングには対応していない。本論文で提案するシステムは、最新の音声合成アルゴリズムの持つ品質の可能性を効率的に引き出すチューニング作業を誰にでも可能とするものである。

2.2 社外の関連動向

音声合成チューニング・ツールは他社からも提供されている。例えば Nuance 社 PromptSculptor [6]、Loquendo 社 TTS Director [7] がその例である。これらのツールは公開されている仕様から判断する限り、1台のコンピュータ上でチューニング・ツールと音声合成エンジンが共に動作するものと思われる。一方で本論文では負荷を分散してチューニング作業の効率化を図るためにクライアント・サーバー・モデルを用いる。

NEC 社の KoeLabo プロジェクト [8] では、専門知識の乏しい一般ユーザーのための合成音声チューニング・ツールを提供している。このプロジェクトではアニメのキャラクターの声で音声合成できる Web アプリケーションを、ユーザー登録した一般ユーザーに制限付きで無料公開している。しかし、このインターフェースは短い1文ずつでしか編集することができないという意味で小規模な作業しか支援していない。本論文の目的は数百といった多数の文についての大規模チューニング作業も支援できるツールである。

この NEC 社のプロジェクトに影響を与えていると考えられるのは、Web 2.0 時代の1つの特徴であると一般にいられている Consumer-Generated Media (CGM) の流行である。CGM の有名な例の1つが「初音ミク [9]」である。これは Yamaha 社の歌唱合成技術 [10] をライセンス供与されたクリプトン・フューチャー・メディア社の VOCALOID2 製品の音声データ [11] に付けられた名前である。グラフィカル・ユーザー・インターフェース (GUI) によってユーザーの創作意欲を刺激し、CGM のマンパワーを発揮しやすくなったのが初音ミクの1つの成功要因と考えられる。音声合成に関して誰でもチューニングできるインターフェースを開発し、CGM のマンパワーを音声合成に利用するのが本論文で提案する

*1 コーパス：統計モデルの機械学習に用いる例文データ。

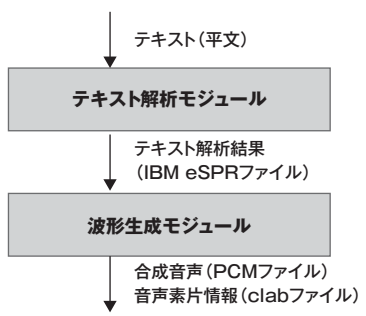


図 1. 従来のシステム構成

システムの狙いでもある。

3. 合成音声チューニングの課題

IBM の音声合成システムは、図 1 のような構成となっている。入力されたテキストに対して、まずテキスト解析モジュールが単語や句の境界、品詞、読み、アクセントを推定する。従来、この推定結果は eSPR フォーマットで保存されていた。次に波形生成モジュールがこのデータに従って音声素片の選択と接続を行って、合成音声を生成する。この際、合成音声波形データで出力されるだけでなく、選択された音声素片や推定したピッチや継続時間長などの実行情報が clab (core label) ファイルと呼ばれる独自データフォーマットで出力される。このシステムには、以下に示す問題点がある。

- (1) 音声合成する際にテキスト解析と波形生成のそれぞれで誤りが生じることがあった。
- (2) テキスト解析と波形生成のそれぞれの出力が独自フォーマットであり、それらの誤りの修正には熟練を要した。
- (3) イントネーションのチューニングを行う際、SWG の TTS Tuner では clab ファイルの表示が可能であったが、グラフィカルにその情報を編集できるものではなかった。
- (4) 波形生成モジュールの計算量が多く、計算機にかかる負荷が大きかった。またこの負荷により速やかなチューニング作業が妨げられてしまうことがあった。

4. 合成音声チューニングの提案システム

本章では提案システムに実現したアイデアをまず説明した後に、システムの詳細を述べる。

4.1 提案システムの主な特長

- (1) 音声合成機能をテキスト解析、波形生成、チューニ

- ング用クライアントに分離し、新たに定義した通信プロトコルで連携させる。
- (2) テキスト解析モジュールと波形生成モジュールはそれぞれサーバーで実行し、サーバー上の豊富な計算機資源を利用する。
- (3) 複数のユーザーが同時にサーバーに接続して作業を行うことができるように、通信プロトコルおよびサーバーを設計する。
- (4) チューニング用クライアントは Flex で開発し、Web ブラウザー上で実行する。
- (5) クライアントではテキスト解析や波形生成の結果を GUI 上で修正し、新たに定義した XML 形式でその中間結果を保存可能とする。
- (6) ユーザーが、あるテキストに関するサーバーの処理結果を待たずに、別のテキストのチューニング作業を進めることができるように、通信プロトコルと GUI を設計する。

4.2 システム構成

システムは新たに、次の要素から構成するように再設計する (図 2)。

- (1) 通信プロトコル：新たに定義した XML 形式データの授受を中心とした、クライアントとサーバー間の通信プロトコル。
- (2) テキスト解析サーバー：受け取ったテキストの解析

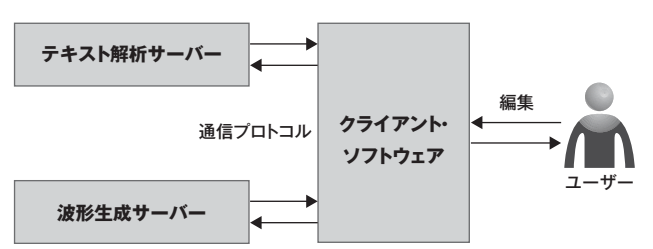


図 2. 新システム構成

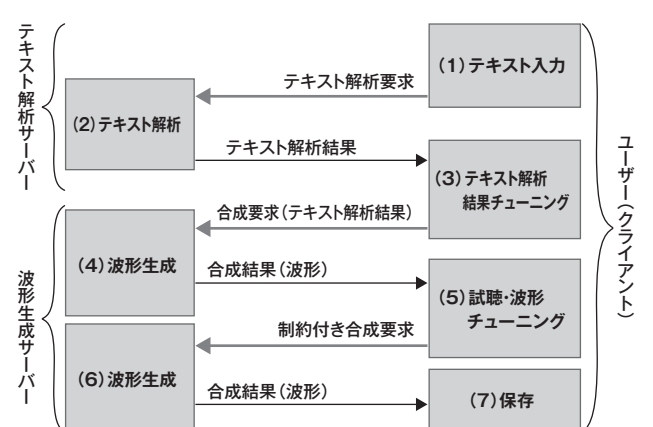


図 3. 提案する典型的な処理フロー

結果を返す。

- (3) 波形生成サーバー：テキスト解析結果を受け取り、合成音声を生成して返す。
- (4) クライアント：ユーザーがチューニング作業を行うことができる GUI を提供する。

これらの要素を用いた典型的な処理フローは次の通りである（図 3）。すなわち、(1) ユーザーが音声を作成したいテキストを入力する。(2) テキスト解析サーバーが解析を行う。(3) ユーザーはその結果を確認・編集する。(4) テキスト解析結果に基づいて波形生成サーバーが合成音声の波形を生成する。(5) ユーザーは合成音声を試聴し、これをチューニングするための制約条件を指定する。(6) 先のテキスト解析結果と制約条件に基づいて波形生成サーバーが再度波形を生成する。(7) ユーザーは完成した波形をファイルに保存する。以下の節では構成要素のそれぞれについて詳説する。

4.3 通信プロトコル

通信内容の中心となる情報は、eSPR 相当のテキスト解析結果、clab 相当の実行情報、WAV ファイルである。このうち WAV ファイルは base64 エンコーディングを行ったテキスト形式で送受信を行う。それ以外の 2 つの情報には XML 形式のデータ・フォーマットを今回新たに定義した。テキスト解析結果の例を図 4 に掲載する。文、イントネーション句、アクセント句、単語、音節（モーラ）、音素の階層構造を表現する内容になっている。通信プロトコルでは複数ユーザーのサーバーへの同時接続や、タスクのキャンセルを実現するために、各通信の際にユーザー名、コマンド番号も併せて通信する。

4.4 テキスト解析サーバー

従来から開発されていたテキスト解析モジュールは、C++ で書かれたコマンドライン・プログラムであり、

```

<Sentence>
  <IntonationalPhrase>
    <Punctuation>comma</Punctuation>
    <PauseDur>50</PauseDur>
    <AccentPhrase>
      <AP_POS>Mesi</AP_POS>
      <Word>
        <Word_POS>Mesi</Word_POS>
        <Grapheme> 音声 </Grapheme>
        <Syllable>
          <SyllStress>1</SyllStress>
          <Phoneme>o</Phoneme>
        </Syllable>
      </Word>
    </AccentPhrase>
  </IntonationalPhrase>
</Sentence>

```

図 4. テキスト解析結果 XML の例

eSPR フォーマットでデータを出力するように設計されていた。今回このプログラムを図 4 の形式で出力するように修正する。さらに、このプログラムを制御する Perl スクリプトを作成し、テキスト解析サーバーとする。テキスト解析サーバーは、ソケットによってクライアントと通信を行う。サーバーには通信機能のほかに、複数文を含む入力を文ごとに分割した上で 1 文ずつテキスト解析モジュールに処理をさせる機能、入力文に付加されたアノテーションを一時的に除去した上でテキスト解析モジュールに平文を与え、テキスト解析モジュールの出力が得られた後に、先に除去したアノテーションをこの出力に付加し直す機能が要求される。

4.5 波形生成サーバー

AIX® 上で動作する IBM Research 版波形生成モジュールを中心として、波形生成サーバーを今回作成する。この波形生成モジュールは、コマンドラインから専用コマンドを受け取り、ファイルに波形ファイルを出力するコマンドライン・プログラムとしてすでに開発されたものである。今回、波形生成サーバーは Perl の POE (Perl

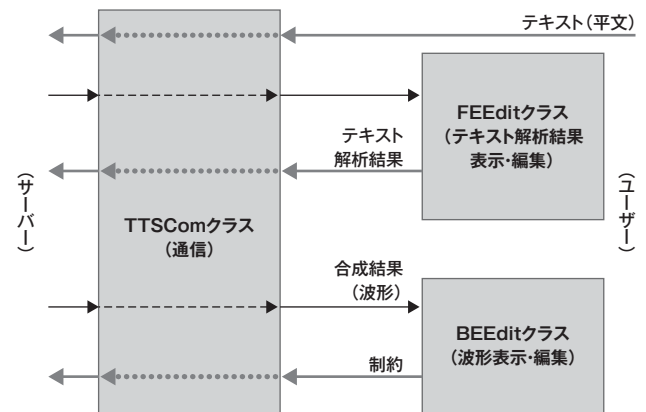


図 5. クライアントの主要コンポーネント

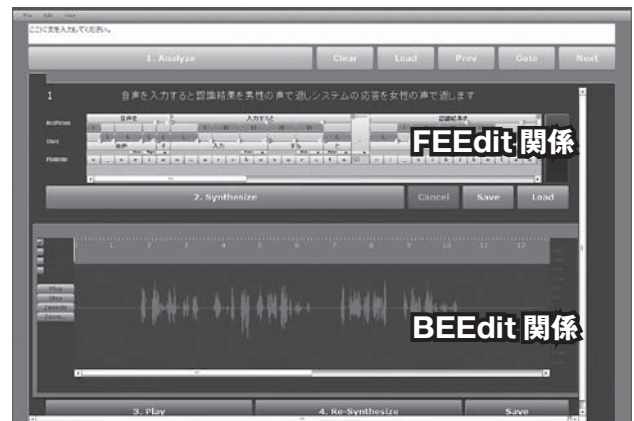


図 6. クライアント GUI



図 7. 文構造を視覚的に表現する FEEdit

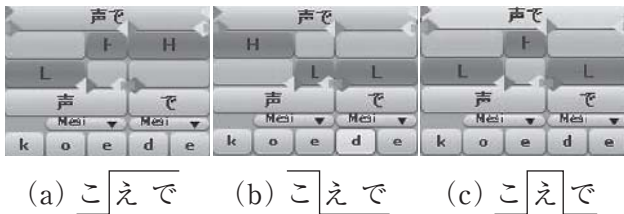


図 8. 句のアクセントを視覚的に表現

表 1. 実験 1 結果

セッション	ツール	修正文数	アクセント精度	句境界精度
1	旧	9	95.6%	96.6%
2	新	16	97.2%	97.7%
3	旧	16	97.8%	96.4%
4	新	29	97.5%	96.0%

Object Environment) モジュール [12] を用いて作成する。POE モジュールは、ファイル入出力やソケット通信などのイベントをトリガーとして処理を行うイベント駆動プログラミングを Perl で実現する拡張モジュールである。波形生成サーバーは、クライアントからソケット通信によって処理要求を受け取り、それを変換して波形生成モジュールの入力へと流し込む。波形生成サーバーは、合成音声の生成終了を検出すると、波形データを WAV ファイルに、clab ファイルを XML 形式に変換した上で、要求元クライアントに対して送信する。

4.6 クライアント

クライアントは、Flex を用いてまったく新しく開発した。開発言語に Flex を選択した理由は GUI 開発に適していることと、プラットフォームに依存せず Web ブラウザー上で利用が可能であることである。

GUI 開発に当たっては、直感的な操作を実現することと、オープンソース化も考慮に入れ明快なクラス構造を心掛けた。最も中心的な役割を果たすクラスは、テキスト解析結果の表示・編集を実現する FEEdit クラス、波形の表示・編集を実現する BEEdit クラス、サーバーとの通信を管理する TTSCom クラスである (図 5)。典型的な作業手順において、操作する対象が画面上から下に向かって順番に移っていくように、FEEdit クラスのボタン類を画面上方に、BEEdit クラスのボタン類を画面下方に配置した (図 6)。複数文に関する作業を並列して進められるように、文をタブに対応付け複数タブを平行して扱えるインターフェースとした。FEEdit クラスでは、テキスト解析の結果である文構造がそのまま

視覚的にも階層的に表示されるようにした (図 7)。また、アクセントを視覚的に表現し、1 クリックで句のアクセントを変更できるようにした (図 8)。BEEdit クラスでは、音声素片のピッチや継続時間長を、ドラッグ操作によって編集することができるようにした。この操作結果は周波数やミリ秒を単位とした物理的な制約条件として記憶され、次の音声合成の際に波形生成サーバーに送信されて使用される。

5. 実験

本章では、本システムの効果を 2 つの利用例と 2 つの実験結果によって示す。

5.1 利用例 1 : e-Learning 教材の作成

弊社テクニカル・リーダーシップ・オフィス (TLO) は 2009 年 4 月 24 日、ITLMC 「IBM 標準となったアジャイル開発」チームによって開発された e-Learning 教材「プロダクトオーナーの基礎」を IBM 社内で公開した。これはパワーポイント資料とそのスクリプトを読み上げるナレーションを組み合わせたコンテンツであるが、このナレーションには IBM 音声合成が実験的に使用されている。コンテンツは 3 時限から構成されており、ナレーションは合計文数で 500 以上、総時間で 2 時間以上に及んでいる。この大量の合成音声の作成には、まだ開発中ではあったが提案システムが使用された。合成音声の一部は第一著者が作成したが、ほかの部分は音声合成の専門家ではないメンバーによって作成された。システムが不完全だったことと合成する分量が多かったこと

もあり決して簡単な作業ではなかったが、提案システムが大規模なコンテンツ開発に利用可能であることを示すことができた事例といえる。

5.2 利用例 2：音声コーパスの修正

本論文で構築したクライアントの一部をそのまま利用し、音声コーパス修正のためのツールを開発した。音声合成用の音声コーパスは、声優が発声した音声の波形データと、その発声内容をテキスト解析した結果を対応付けたデータになっている。正確な音声コーパスを利用することが、高精度な統計モデルを機械学習するために重要である。音声を聴いて発声内容と一致するようにテキスト解析結果データを手作業で修正するのが、音声コーパス修正作業である。筆者らは今回作成したツールを用いて実際に 8,851 文の分量のある音声コーパスについてすべて音声を聴取し、テキスト解析結果の修正を行った。この音声コーパスは、すでにその大部分につき別のツールを用いてアクセントの修正を行ったものであったが、今回の作業によってさらに約 2.77% のモーラにつきアクセントを修正、約 14.5% のアクセント句境界を単語境界に修正することになった。このツールを用いることで大規模の音声コーパス修正を短期間に一人で行うことができたため、修正内容の一貫性も向上したことが期待できる。

5.3 実験 1：テキスト修正効率

利用例 2 で使用した新ツールによるテキスト解析結果の修正効率向上を実験によって検証する。被験者は、アクセントの聴取やテキスト解析結果の修正の経験がまったくない 1 名のユーザーとした。比較対象の旧ツールとしては通常のテキスト・エディター（Windows® Notepad）を用いる。実験では 1 セッション 15 分間で被験者が修正できる音声コーパスの分量とその精度を測定した。第一著者があらかじめ作成したコーパスを正解

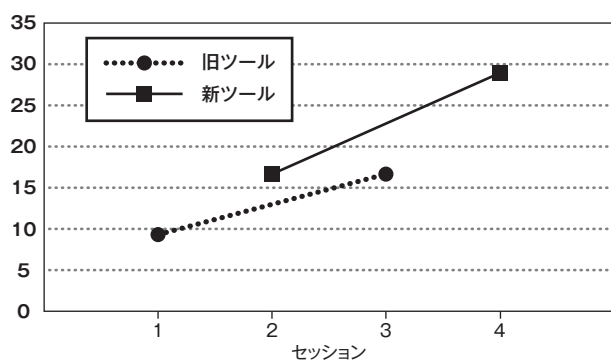


図 9. 実験 1 結果

表 2. 実験 2 結果

	自然	やや不自然	不自然
旧モデル	69.9%	13.1%	17.0%
新モデル	77.8%	12.1%	10.1%

とし、アクセント精度としてモーラ単位でのアクセント一致率および、句境界精度として単語境界からアクセント句境界を特定する精度（F 値）を測定した。セッションの順番は（1）旧（2）新（3）旧（4）新とした。

結果を表 1 および図 9 に示す。若干の違いはあるが両ツールで精度は同程度であった一方、図 9 に見るように作業効率には違いを見いだすことができた。この被験者は音声アクセントの聴取経験がなかったために、音声を繰り返し聴き直す時間が作業時間の大半を占めていた。セッションが進むにつれアクセントの判断が速くなり、ツールに無関係に作業効率が改善されている。しかし、新ツールでは 1 回目のセッションで、旧ツールの 2 回目のセッションと同じ分量の修正を達成し、さらに 2 回目のセッションでは 1 回目のほぼ倍を修正することができた。以上から新ツールの作業効率が良いことが分かる。

5.4 実験 2：ピッチ・モデル^{※2}精度

実験 2 では、利用例 2 で修正した音声コーパスによるピッチ・モデルの精度向上を測定する。100 文のテキストにつき、新しいピッチ・モデルと従来のピッチ・モデルの双方を用いて合成音声を生成した。新旧のピッチ・モデルは、利用例 2 の修正の後の音声コーパスを学習に使用したか、修正前の音声コーパスを使用したかだけが異なっている。ランダム順にして合成音声は新旧どちらで作成したか分からなくした状態で、各合成音声を聴取してアクセントの正確さを評価した。評価は（1）自然、（2）やや不自然、（3）不自然の 3 段階で行った。ピッチ・モデルの正確さを純粋に評価するために、モデルで推定したピッチを最終的な合成音声に忠実に利用するモードで合成音声を生成した。

結果を表 2 に示す。新ピッチ・モデルは旧ピッチ・モデルに対して 7.9 ポイント「自然」が増え、「不自然」が 6.9 ポイント減少している。このように今回作成したツールが音声コーパス修正を効率化し、ピッチ・モデル精度でも音声合成システムの改善に貢献したことが分かる。

※2 ピッチ・モデル：テキスト解析結果に基づいて、合成音声に用いるピッチ（音の高さ）を推定する統計モデル。

6. おわりに

音声合成機能をクライアントとサーバーに分離し、その間の通信プロトコルを XML 形式で定義して、直感的な GUI クライアントを Flex によって実装した。システムの有効性は大規模な e-Learning 教材の作成と大規模な音声コーパス修正という利用事例と、テキスト修正効率とピッチ・モデルの精度向上の実験によって示した。

現在明らかになっている課題として、波形編集機能の実装が不完全であること、波形生成サーバーの速度が遅いこと、Flex が完全なマルチ・タスク処理を実現していないためファイルの読み込みや通信中に処理が中断されたり通信に失敗したりすることがあることなどが挙げられる。

謝辞

豊橋技術科学大学荻山将成さんにはクライアントを開発していただきました。日本 IBM（当時）の藤田雅之さんには多大な手間をかけて開発中のシステムをお使いいただきました。あらためて深謝いたします。

参考文献

- [1] Adobe Flex 3, <http://www.adobe.com/jp/products/flex/> (2010.5.19).
- [2] IBM ホームページ・リーダー Windows 版 ver3.04, http://www.ibm.com/jp/accessibility/solution_offerings/hpr/index.html (2010.5.19).
- [3] J. F. Pitrelli, et al., "The IBM expressive text-to-speech synthesis system for American English," IEEE Trans. on Audio, Speech and Language Processing, vol. 14, no. 4, pp. 1099-1108, (2006.7).
- [4] T. Saito, et al., "ProTALKER: a Japanese Text-To-Speech System for Personal Computers," Tech. Rep. RT0110, IBM TRL Research Report, (1995.6).
- [5] R. Bakis, et al., "System for tuning synthesized speech," 2007, United States Patent Application Publication, US 2008/0167875 A1.
- [6] Nuance PromptSculptor, <http://www.nuance.com/promptsculptor/> (2010.5.19).
- [7] Loquendo TTS Editor, http://www.loquendo.com/en/technology/tts_director.htm (2010.5.19).
- [8] 思い通りにキャラがしゃべる KoeLabo, <http://voice.biglobe.ne.jp/index> (2010.5.19).
- [9] VOCALOID2 初音ミク, <http://www.crypton.co.jp/mp/pages/prod/vocaloid/cv01.jsp> (2010.5.19).
- [10] H. Kenmochi and H. Ohshita, "VOCALOID - commercial singing synthesizer based on sample concatenation," Proc. Interspeech, pp. 4009-4010. (2007.8).
- [11] クリプトン・フューチャー・メディア VOCALOID2, <http://www.crypton.co.jp/mp/pages/prod/vocaloid/> (2010.5.19).
- [12] POE: Perl Objective Environment, <http://poe.perl.org/> (2010.5.19).

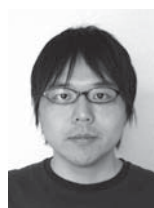


日本アイ・ビー・エム株式会社
東京基礎研究所
主任研究員

立花 隆輝 Ryuki Tachibana

[プロフィール]

1998 年 日本 IBM 入社。以来東京基礎研究所において音楽電子透かしと音声合成の研究に従事。電子情報通信学会, 日本音響学会各会員。博士 (工学)。



日本アイ・ビー・エム株式会社
東京基礎研究所
主任研究員

長野 徹 Tohru Nagano

[プロフィール]

1998 年 日本 IBM 入社。以来東京基礎研究所にて自然言語処理および音声言語処理の研究に従事。情報処理学会, 日本音響学会各会員。



日本アイ・ビー・エム株式会社
東京基礎研究所
主席研究員

西村 雅史 Masafumi Nishimura

[プロフィール]

1983 年 日本 IBM 入社。以来東京基礎研究所において音声認識・音声合成の研究に従事。1998 年 情報処理学会山下記念研究賞, 1999 年 日本音響学会技術開発賞各受賞。電子情報通信学会英文論文誌編集委員会幹事, シニア会員。IEEE, 日本音響学会, 情報処理学会各会員。音声技術担当。博士 (工学)。