

# 子供の負担を考慮した子供音声収集システム

河原 誠斗<sup>1</sup> 堤 修平<sup>1</sup> 山本 大介<sup>1</sup> 高橋 直久<sup>1</sup>

**概要：**子供音声認識の技術はゲームや、教育分野での大きな需要が期待されている。しかし、子供音声の認識率は成人音声のものと比べ低い。さらに、子供音声データベースの作成は幼い子供には負担がかかり、音声収録が困難であるという問題点が挙げられる。また、我々は音声インタラクション構築ツールキット MMDAgent の研究を行っている。MMDAgent は FST ファイルと呼ばれるファイルを編集し、対話内容を記述することで音声対話を実現している。本研究では、MMDAgent を用いた、子供の負担を考慮した音声収集システムの実現を目指す。提案手法では、ゲーミフィケーションによって子供の負担を軽減するさらに、作成したプロトタイプシステムに対する評価実験を行い、提案システムの有用性を評価した。

## A Voice Collecting System without much Stress for Children

MASATO KAWAHARA<sup>1</sup> SHUHEI TSUTSUMI<sup>1</sup> DAISUKE YAMOMOTO<sup>1</sup> NAOHISA TAKAHASHI<sup>1</sup>

### 1. はじめに

近年、ポータルサービス、カーナビゲーションシステム、パソコン操作補助などに音声認識技術の応用が進められている。これらは、主に成人音声を認識することを前提として開発されているが、最近では、音声認識技術の利用者層のさらなる拡大を目指し、子供音声や高齢者音声の対応も進められている。特に子供音声認識はゲームや教育分野での大きな需要が期待される。しかし、子供音声データベースの作成は幼い子供には負担がかかり、音声収録が困難であるという問題点がある。

また、本研究室では音声インタラクション構築ツールキット MMDAgent を用いて研究を行っている。MMDAgent[1][2] は、名古屋工業大学国際音声技術研究所で開発された音声対話システムであり、画面上に 3D キャラクタが表示され、ユーザは表示された秘書エージェントと会話ができる。名古屋工業大学正門前に設置された双方向音声案内デジタルサイネージなどにも応用されている。FST ファイルと呼ばれるファイルを編集し、対話シナリオを記述することで音声対話を実現している。また、ユーザ辞書ファイルを編集することでシステム辞書だけでは認識できない単語も認識可能にすることができる。FST ファイルとユー

ザ辞書ファイルの詳細については、2章で述べる。

本研究では、MMDAgent を用いたゲーミフィケーションによる音声収録で、子供の負担を軽減する手法を提案する。

### 2. 音声インタラクション構築ツールキット MMDAgent

#### 2.1 MMDAgent の特徴

MMDAgent は主に、音声認識部、音声合成部、対話管理部、3D モデルの描画と物理演算などモーションを管理するエージェント部を密統合したシステムである。音声認識部には汎用大語彙連続音声認識エンジン Julius を、音声合成部には日本語テキスト音声合成エンジン Open JTalk を、3D モデルには MikuMikuDance 形式を、また、自然法則に従ったリアルな動きを与えるため、物理演算エンジンとして Bullet Physics を採用している。

#### 2.2 FST ファイル

FST ファイルは状態遷移を記述したテキストファイルである。遷移元の状態遷移番号と遷移先の状態遷移番号、状態遷移条件、状態遷移時に実行されるコマンドの 4 つ組で構成されている。図 1 のように、この 4 つの要素を 1 行に空白かタブ文字で区切って記述することで対話の状態遷移を表す。MMDAgent では内部メッセージが発行されてお

<sup>1</sup> 名古屋工業大学大学院工学研究科 Graduate School of Engineering, Nagoya Institute of Technology

り、イベントキューを介したメッセージのやり取りによって対話管理が行われている。イベントメッセージは状態遷移条件を表し、コマンドメッセージは実行されるコマンドを表す。

```
1 10 RECOG_EVENT_STOP|こんにちは <eps>
10 11 <eps> SYNTH_START|mei|mei_voice_normal|こんにちは
11 1 SYNTH_EVENT_STOP|mei <eps>
```

図 1 FST ファイルの一例

### 2.3 ユーザ辞書ファイル

ユーザ辞書ファイルに単語を登録することで任意の単語を音声認識可能にできる。また、登録する単語の認識されやすさを変更することができる。ユーザ辞書ファイルに単語を登録する際、図 2 のように 5 つ組を記述する。第 1 フィールドは単語の出現を表す文字列を指定する。システム辞書内にある単語の中で似た単語の第 1 フィールドを指定する。音声認識時には、ここで指定された単語の出現パターンがそのままこの新単語に適用される。図 2 の 1 行目の例では「鶴舞」は地名であり、システム辞書内にある「東京：トーキョー：東京：515」を指定することで出現を定義している。分からない場合は「unk」と記述して、下記の第 2 フィールドの補正值を変えることで手動で調整することもできる。第 2 フィールドには認識補正值を設定する。@の後の数字が補正值で、大きくするほど認識されやすくなる。登録する単語が認識されにくい場合はこの数字を大きくすることで認識されやすくなる。第 3 フィールドにはその単語の登録情報を、表記:発音:基本形の順で記述する。数字は第 1 フィールドと同じものを記述する。また第 1 フィールド同様、分からない場合は「unk」と記述する。第 4 フィールドには認識結果として出てくる単語を記述する。第 5 フィールドには読みを表す音素列を記述する。

```
1 東京:トーキョー:東京:515 @1.0 鶴舞:ツルマイ:鶴舞:515 [鶴舞] tsurumai
2 花子:ハナコ:花子:513 @2.0 メイ:メイ:メイ:513 [メイ] mei
3 花子:ハナコ:花子:513 @2.0 メイ:メイ:メイ:513 [メイ] me:
4 <unk> @0.0 <unk> [^0~] heroo:
```

図 2 ユーザ辞書ファイルの一例

## 3. 提案システムの概要

### 3.1 提案システムの構成

提案システムの構成を以下に示す。提案システムの構成図は図 3 のようになる。提案システムでは三択問題を用いて、子供の負担を軽減する。

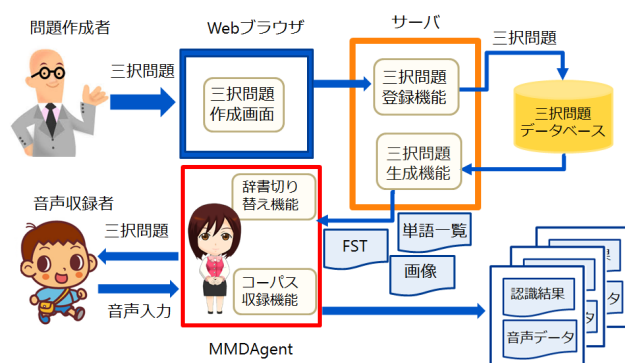


図 3 システム構成図

**三択問題登録機能** Web ブラウザに入力された三択問題を三択問題データベースに追加する。

**三択問題生成機能** MMDAgent で三択問題を行うために必要なファイルを生成する。

**辞書切り替え機能** 三択問題の問いが変わるたびにユーザ辞書ファイルの内容を書き換え、再読み込みをする。これにより、三択問題の選択肢の単語が一時的に認識されやすくなり、子供音声での音声対話の際に生じるストレスを軽減できる。

**コーパス収録機能** 音声入力が行われるたびにフォルダを生成し、認識結果と音声データを合わせて保存する。

### 3.2 提案システムの動作

システムの動作を以下に示す。

**三択問題の登録** 問題作成者は Web ブラウザの入力フォームに三択問題を入力する。入力された三択問題は、三択問題登録機能により、三択問題データベースに追加される。

**三択問題のファイル生成** 三択問題生成機能により、三択問題データベースの情報から MMDAgent に必要な FST ファイル、選択肢単語一覧、三択問題の選択肢を MMDAgent の画面上に表示するための画像を生成する。

**MMDAgent を用いた三択問題** 音声収録者となる子供に MMDAgent を用いた三択問題を行ってもらう。

**コーパス収録** MMDAgent に入力された音声は、コーパス収録機能により、認識結果と合わせて保存される。

## 4. 提案システムの実現法

提案システムは三択問題登録機能、三択問題生成機能、辞書切り替え機能、コーパス収録機能からなる。それぞれの実現法を以下に示す。

### 4.1 三択問題登録機能

HTML を用いて Web ブラウザから三択問題の登録を行う。三択問題は図 4 のような入力フォームから登録する。

問題文は 101 文字以上、選択肢は 8 文字以上入力できなくなっている。入力された「問題文」、「正答」、「誤答 1」、「誤答 2」は Java Servlet へ送信され、データベースに格納される。Java Servlet では、入力されたデータに入力漏れがないか判定を行った後、データベースに格納する。データベースの構成を表 1 に示す。

## 三択クイズ登録

▼問題文を入力してください。(100字以内)

問題文

▼問題の答えを入力してください。(10字以内)

答え

▼誤答を 2 つ入力してください。(10字以内)

誤答 1

誤答 2

送信

リセット

▼登録済みデータの確認

GO

▼登録済みデータからファイル生成

Download

図 4 三択問題登録画面

表 1 三択問題データベース

入力フォーム	カラム名	型	備考
	ID	INT	AUTO.INCREMENT
問題文	QUESTION	VARCHAR(256)	NOT NULL
正答	MARU	VARCHAR(20)	NOT NULL
誤答 1	BATSU1	VARCHAR(20)	NOT NULL
誤答 2	BATSU2	VARCHAR(20)	NOT NULL

## 4.2 三択問題登録機能

データベースに保存された三択問題をもとに、選択肢を表示するための画像ファイル、FST ファイル、選択肢単語一覧を生成する。

### 4.2.1 画像ファイル生成

三択問題を行う際、MMDAgent の画面上に三択問題の問いの選択肢を表示する。これのもととなる画像を生成する。三択問題データベースに保存されている正答、誤答 1、誤答 2 の情報から画像を生成する。1 単語につき 1 つフォルダを生成し、画像と保存する。

三択問題の選択肢を表示する際、図 5 のように選択肢の単語を雲の形で表示するための PMD ファイルも画像と同じフォルダに保存する。



図 5 三択問題

### 4.2.2 FST ファイル生成

データベースに保存された三択問題情報をもとに、FST ファイルに対話スクリプトを書き出す。生成する FST スクリプトの状態遷移の流れを図 6 に示す。三択クイズは 10 問ずつパックわけされる。三択問題全体の FST は次のように状態遷移する。

- ステップ 1 初期状態からクイズと発話し、パック選択の状態へ遷移する。
- ステップ 2 キー入力または、音声によりパック選択をし、三択問題が始まる。
- ステップ 3 三択問題に正解すると次の問題へ遷移する。不正解の場合は次の問題へ遷移せず、もう一度同じ問題を読み上げる。
- ステップ 4 ステップ 3 を繰り返し、10 問目の問題に正解すると初期状態へ戻る。

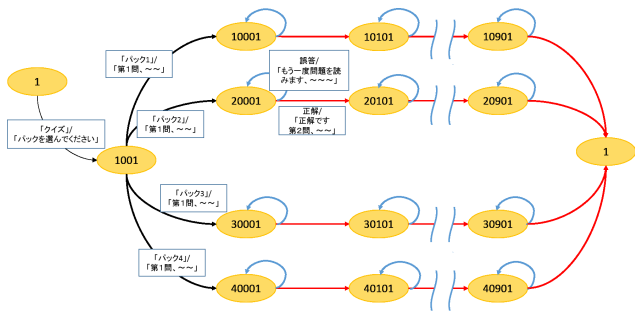


図 6 三択問題全体の状態遷移の流れ

MMDAgent の FST スクリプトには、モデル、モーション、移動/回転、音楽/画像、カメラ/証明、音声認識、音声合成などのコマンドメッセージ・イベントメッセージが用意されている。FST からユーザ辞書ファイルの切り替えを行うために、「CHANGE\_DIC|question id」というコマンドメッセージを設定する。辞書切り替え機能の詳細については 4.3 節に示す。また、選択肢の単語が表示する位置をランダムにするために、FST スクリプトの乱数を用いる。図 7 に三択問題 1 問分の状態遷移を示す。次のようなステップで状態遷移する。

ステップ 1 クイズが始まると、問題を読み上げる。

ステップ 2 コマンドメッセージ「CHANGE\_DIC |question id」を発行する。(question id) には三択問題データベースに格納されている問題番号を記述する。

ステップ 3 乱数を用い、正解の選択肢の位置を三箇所からランダムに表示し、2つの誤答を残りの二箇所に表示する。

ステップ 4 解答受付状態になり、音声収録者からの発話を待つ。

ステップ 5 正解を認識した場合、選択肢を削除し次の問題へ遷移する。不正解を認識した場合、問題文をもう一度読み上げ再び解答受付状態へ戻る。「もう一回」と認識した場合、不正解を認識したときと同様に問題文をもう一度読み上げ解答受付状態へ戻る。「終了」または、「おしまい」と認識した場合、三択問題を終了し、初期状態へ遷移する。

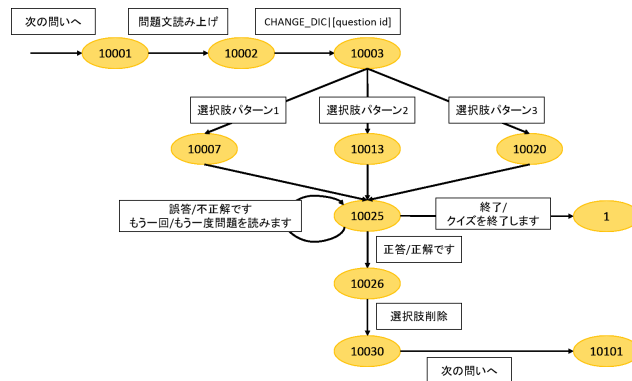


図 7 三択問題 1 問分の状態遷移

#### 4.2.3 選択単語一覧生成

データベースに保存されている正答、誤答 1、誤答 2 の情報から選択肢の単語の一覧を生成する。このとき、図 8 のように、単語はユーザ辞書ファイルの記述形式でテキストファイルに記述する。第 1 フィールドと第 3 フィールドは全て「unk」とし、選択肢の単語の認識補正値は@5.0 とした。また、選択肢の単語のほかに、「もう一回」「おしまい」「終了」の 3 単語も初めの 3 行にユーザ辞書ファイルの記述形式で記述する。生成された単語一覧は 4.3 節で示す辞書切り替え機能で利用される。

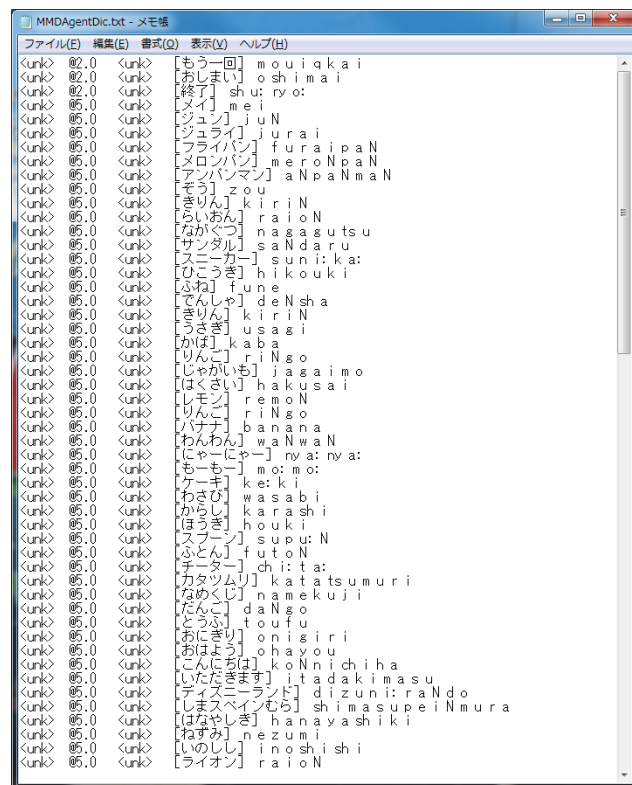


図 8 選択肢単語一覧

### 4.3 辞書切り替え機能

ユーザ辞書ファイルを切り替える機能を MMDAgent のプラグインとして作成する。

図 9 に示すように、三択問題の問いが変わるたびにユーザ辞書ファイルの内容を変更し、再読み込みを行う。以下に示す手順でユーザ辞書ファイルを切り替える。

- 手順 1 三択問題の問いが切り替わると FST スクリプトでコマンド「CHANGE\_DIC|(question id)」を発行する。
- 手順 2 コマンドメッセージから問題番号を取り出す。
- 手順 3 4.2.3 節で述べた方法で生成された選択肢の単語一覧から該当する 3 行を検出する。
- 手順 4 単語一覧の検出した 3 行と初めの 3 行の内容を抽出する。
- 手順 5 ユーザ辞書ファイルを抽出した内容に書き換える。
- 手順 6 ユーザ辞書ファイルを再読み込みする。

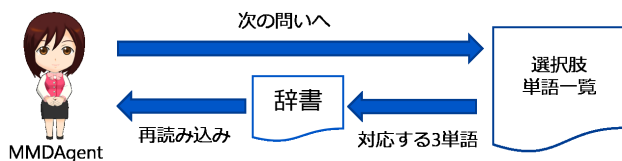


図 9 辞書切り替え

### 4.4 コーパス収録機能

1 発話分の音声と MMDAgent の認識結果を合わせて保存する機能を MMDAgent のプラグインとして作成する。この機能を以下の手順で実現する。

- 手順 1 収集する音声を保存するフォルダを生成しておく。
- 手順 2 マイクからの音声入力を Julius を用いて音声区画検出を行い、1 発話分の音声として分割し、録音する。
- 手順 3 FST スクリプトでコマンドメッセージ「RECOG\_EVENT\_STOP」を受け取ると、生成しておいたフォルダ内に新しくフォルダを生成する。同一名のフォルダが生成されないように、フォルダ名は生成されたときの時刻を用いる。
- 手順 4 認識結果をテキストファイルに書き出し、生成したフォルダ内に保存する。
- 手順 5 録音した 1 発話分の音声も同フォルダ内に保存する。

## 5. プロトタイプシステム

プロトタイプシステムは、MMDAgent に実装した辞書切り替え機能、コーパス収録機能と三択問題登録機能、三択問題生成機能を持ったプログラムで構成される。三択問題登録機能、三択問題生成機能は Windows 環境で Tomcat[9] の Web サーバを使用し、Java 言語 [10] を用いて実装した。辞書切り替え機能、コーパス収録機能は C++ を用いて実装した。MMDAgent は C++ で実装されており、公開されているバージョン 1.5 を使用している。開発環境は Java の実装に Eclipse[11] を、C++ の実装に Visual Studio[12] を用いた。データベースには Java で実装された HSQLDB(HyperSQL DataBase)[13] を用いる。

### 5.1 三択問題登録機能

三択問題を登録する Web ページにアクセスすると図 10 の画面が表示される。入力フォームにしたがって、三択問題を入力する。このとき問題文は 100 文字以内、選択肢の単語はそれぞれひらがなまたは、カタカナで 8 文字以内という制限がある。

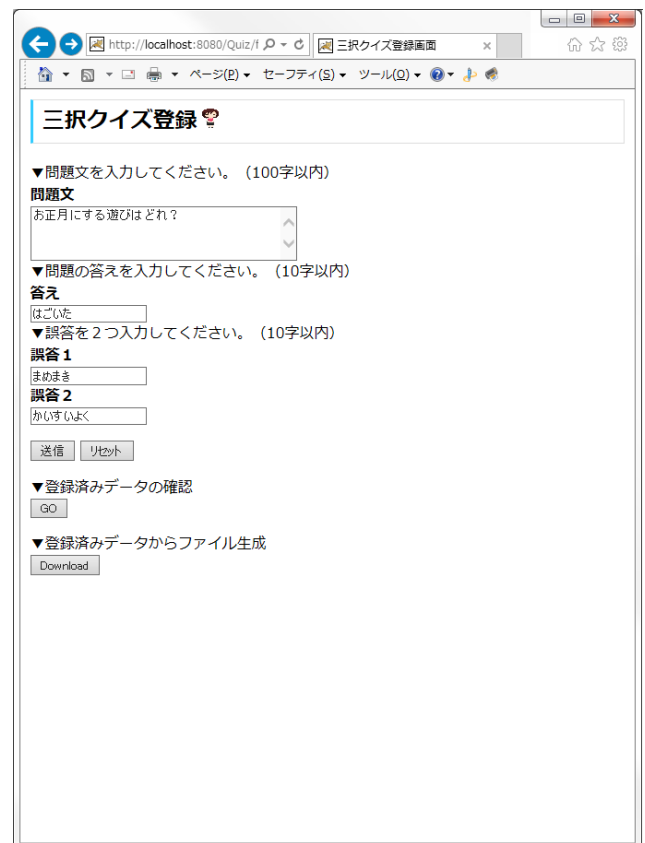


図 10 三択問題登録画面

すべての項目に入力した後、「送信」ボタンを押すと登録内容表示画面が表示され、登録した内容とすでに登録されている内容が確認できる。入力を初めからやり直すときは「リセット」ボタンを押す。



三択問題登録画面の「GO」ボタンを押すと三択問題データベースに登録されている内容を確認することができる。

## 5.2 三択問題生成機能

三択問題登録画面の「Download」ボタンを押すと、三択問題データベースに格納されている情報から選択肢単語一覧、FST ファイル、選択肢を表示するための画像ファイルがダウンロードできる。また、ダウンロードと同時に MMDAgent が読み込む各ファイルが自動的に更新される。

## 5.3 音声収録

MMDAgent を起動し、音声収録者は三択問題を行う。このとき、辞書切り替え機能により、問いごとにユーザ辞書ファイルが更新され、選択肢の単語の認識率が一時的に上がった状態で三択問題を行うことができる。また、コーパス収録機能によって入力された音声は1発話ごとに区切られ、MMDAgent の認識結果と合わせて保存されていく。

## 6. 評価実験

### 6.1 実験1

#### 6.1.1 実験の目的

提案システムを利用して音声収録を行い、以下の3点を検証することが目的である。

項目1 子どもの負担を軽減できているか

項目2 子供音声を正しく認識できるか

項目3 コーパスは正しく生成されているか

#### 6.1.2 実験方法

被験者に4.3章で示した辞書切り替え機能を実装していない従来のシステムを用いて音声入力をしてもらい、誤認識の回数を計測する。次に提案システムを用いた音声収録を行ってもらい、その際、誤認識の回数、収録にかかった時間を計測する。その後アンケートに回答してもらい、被験者は6歳男児と10歳女児の二名である。それぞれ、被験者A,Bとする。

#### 6.1.3 実験手順

実験手順は以下のようである。被験者二名に同時に実験を行った。

手順1 被験者に簡単なMMDgentの紹介をする。

手順2 基本的な日常会話の対話シナリオを記述したFSTファイルを用いた従来のシステムのMMDAgentに自由に音声入力をしてもらう。このとき、ユーザ辞書ファイルには何も記述しない。

手順3 誤認識の回数を計測する。

手順4 提案システムを用いたMMDAgentで三択問題を

行ってもらい音声収録をする。

手順5 三択問題の30問を1問ずつ交互に答えてもらう。

手順6 誤認識の回数、収録時間を計測する。

手順7 アンケートに回答してもらう。

手順8 コーパス収録機能によって保存された音声と文字列を1つずつ確認する。

アンケートの内容は表2のとおりである。それぞれ5段階評価を行った。また、アンケートの最後に自由記述欄も用意した。

表2 アンケート項目

番号	評価項目
1	楽しかったか
2	音声は聞き取りやすかったか
3	またやりたいか
4	音声入力に対する抵抗はあったか

#### 6.1.4 結果と考察

従来システムを用いた場合、被験者は表3に示す言葉を発声した。

表3 入力された音声

被験者 A	被験者 B
こんにちは	こんにちは
好きな食べ物は?	おはよう
あっち行って	特技は?
バイバイ	バイバイ

被験者Aは上記の4つの言葉をそれぞれ2~3回発声し、被験者Bは4つの言葉をそれぞれ1回ずつ発声した。このとき、被験者Aは10回の発声のうち全てが正しく認識できず、被験者Bの場合、「こんにちは」と「バイバイ」と発声したときのみ正しく認識された。被験者Bは4回の発声のうち2回が認識できなかったという結果が得られた。

次に、提案システムを用いた場合のアンケートの各項目の評価、誤認識の回数を表4に示す。

表4 アンケート結果

項目	被験者	
	被験者 A	被験者 B
1. 楽しかったか	4	5
2. またやりたいか	5	4
3. 音声は聞き取りやすかったか	4	4
4. 音声入力に対する抵抗はあったか	5	4
5. 誤認識の回数	4回	0回

項目1,2からゲーミフィケーションによる子供の負担の軽減は効果的だったといえる。しかし、「三択問題が簡単す

ぎた」という意見もあった。これは、難易度や問題のジャンルなどで三択問題の傾向を選べるようにすることで解決できると考えられる。

また、項目5から誤認識の割合が従来システムと比べ低くなっている。このことから、辞書切り替え機能により音声認識率を上げることができたと考えられる。

次に、被験者Aの音声を正しく認識できなかった三択問題を以下に示す。

問い 苦い飲み物はどれ？

正答 コーヒー

誤答1 コーラ

誤答2 カルピス

このとき、被験者Aは4回「コーヒー」と発声したが全て「コーラ」と認識された。これは、音素が互いに似ている単語が選択肢として登録されていたためであると考えられる。保存されたコーパスは音声データと認識結果が適切に合わせて保存されていたが、雑音による音声データやその認識結果も保存されていた。雑音は学習データとして不適当であるため、収集したコーパスの中から不適当なデータを削除する機能が必要であると考えられる。

## 6.2 実験2

### 6.2.1 実験の目的

互いに音素の似ている単語がユーザ辞書ファイルに登録されているとき、誤認識が発生しやすくなる。実験2の目的は単語同士の類似度と認識率の関係性を検証することである。

### 6.2.2 実験方法

単語の類似度は、文字列の最小編集距離（レーベンシュタイン距離）を用いて測る。レーベンシュタイン距離は一文字の挿入・削除・置換によって、文字列を別の文字列に変形するのに必要な手順の最小回数として定義されている。ユーザ辞書ファイルに入力する音声と一致する単語と、その単語と音素の似ている単語を1つずつ記述する。実験1で収集した子供音声を用いてMMDAgentに音声入力をし、2単語の音素のレーベンシュタイン距離と子供音声認識率の関係について検証する。実験2では実験1で誤認識が多く発生した単語「コーヒー」で実験を行う。

### 6.2.3 実験手順

実験手順は以下のようである。

- 手順1 ユーザ辞書ファイルに単語「コーヒー」を記述する。
- 手順2 音素が「コーヒー」に似た単語を登録する。
- 手順3 実験1で収集した子供音声を用いてMMDAgentに音声入力を行う。
- 手順4 入力された音声が正しく認識されている確認する。
- 手順5 ユーザ辞書ファイルの内容を変更する。

手順6 手順2～4を繰り返し、レーベンシュタイン距離と子供音声認識率の関係について検証する。

### 6.2.4 結果と考察

実験結果は図11のようになった。

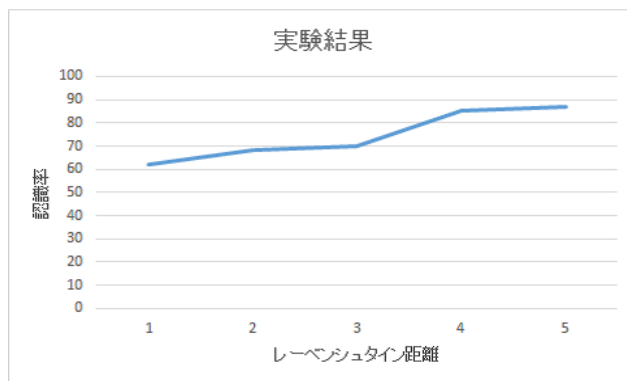


図11 実験結果

レーベンシュタイン距離が1～3のとき認識率は約65%となり、距離が4,5のとき約85%となった。また、文字の編集が削除のみの場合、正しく認識されない傾向にあった。ユーザ辞書ファイルに登録されている単語の音素が似ているとき、音素の文字数が少ない単語が認識されやすいと考えられる。

## 7. 関連研究

### 7.1 小学生音声データベースを用いた子供音声認識の検討

文献[7]は、小学生音声データベースの構築のためにあらかじめ用意した単語セットを子供に読み上げてもらうことで音声を収集している。収集はアニメーションキャラクターとの対話形式で行う。画面に文字とイラスト、アニメーションキャラクターを表示し、ボタンを押して発声する。構築した小学生音声データベースを用いて小学生音響モデルを作成し、実験により認識率の変化の傾向を示している。

本研究とは画面上のキャラクターとの音声対話を用いて音声収集をするという点で関連している。文献[7]では、子供の負担を軽減するために上記のような手法を用いているが、本研究では、ゲーミフィケーションによる音声収録で子供の負担を軽減するという点で異なっている。

### 7.2 実環境における子供音声認識のための音韻モデルおよび教師なし話者適応の評価

文献[8]は、施設の案内や奈良県生駒市の観光案内、周辺の案内情報などを行う音声情報案内エージェントを用いている。ユーザに自由に話しかけることを許すことで、機会に対する人間の自由な発話を収集する。このシステムによって集めた子供音声を用いて高精度な子供音声認識を目指し、その方法を提案している。収集した音声すべてに対して、人手による雑音データのラベル付け、発話内容のテ

キストへの書き起こし, および性別・年齢層・雑音種類・有効/無効のラベル付けを行っている。

本研究とは文献 [7] と同様に画面上のキャラクタとの音声対話を用いて音声収集をするという点で関連している。本研究では, 収集した音声の発話内容を自動でテキストに書き起こしているが, 文献 [8] では, 人手によって様々な情報をラベル付けしているという点が相違点として挙げられる。

## 8. おわりに

本論分では, 音声対話システムである MMDAgent を用いて, ゲームフィクションによって子供の負担を考慮した子供音声収集システムを提案し, その実現方法について述べた。提案システムは, 直接 FST ファイルを編集せずとも, Web ブラウザから三択問題の追加ができるという機能を持つ。また, 子供音声でも認識されやすくするために, 三択問題の選択肢の単語の認識率を一時的に上げる機能を持つ。さらに, 収集した音声を学習データとして扱いやすくするために音声データと認識結果の文字列を合わせて保存する機能を持つ。提案した実現方法をもとに音声収集システムを実装し, 評価実験を行った。子供音声での誤認識の回数が減り, 音声収録の際の子供の負担の軽減がされているという結果が得られた。また, 互いに音素の似た単語を選択肢として登録したとき認識率が低くなるという結果も得られた。今後の課題として, 収集する音声の音素バランスを考慮すること, 子供が飽きないように三択問題を難易度やジャンルでわけることやユーザインタフェースの改良, 学習データとして不適切な音声を検出し削除する機能の追加をすることが挙げられる。

## 謝辞

本研究は JSPS 科研費 26330136, 25700009 および, 科学技術振興機構の助成を受けたものです。この場を借りて, 感謝の意を表します。

## 参考文献

- [1] MMDAgent  
<http://www.mmdagent.jp/>(2016.2.1 参照)
- [2] 李晃伸, 大浦圭一郎, 徳田恵一, 魅力ある音声インタラクションシステムを構築するためのオープンソースツールキット MMDAgent, 電子情報通信学会技術研究報告. NLC, 言語理解とコミュニケーション Vol.111 No.364, pp.159-164, 2011
- [3] Lee Akinobu, and Tatsuya Kawahara, Recent Development of Open-Source Speech Recognition Engine Julius, Proceedings of the APSIPA ASC, pp.131-137, 2009.
- [4] 大浦圭一郎, 酒向慎司, 徳田恵一, 日本語テキスト音声合成システム Open JTalk, 日本音響学会春季講論集, Vol. 1, No.2-7-6, pp. 343-344, 2010.
- [5] MikuMikuDance  
<http://www.geocities.jp/higuchuu4/>(2016.2.1 参照)

- [6] Bullet Physics  
<http://bulletphysics.org/wordpress/>(2016.2.1 参照)
- [7] 小川 厚徳, 山口 義和, 松永 昭一, 小学生音声データベースを用いた子供音声認識の検討, 電子情報通信学会論文誌.SP, 音声, Vol.102 No159, pp.1572-1580, 2004
- [8] 鮫島 充, ランディゴメス, 李 晃伸, 猿渡 洋, 鹿野 清宏, 実環境における子供音声認識のための音韻モデルおよび教師なし話者適応の評価, 情報処理学会論文誌, Vol.47 No.7, 2295-2304, 2006
- [9] Apache Tomcat  
<http://tomcat.apache.org/>(2016.2.1 参照)
- [10] Java  
<http://java.sun.com/>(2016.2.1 参照)
- [11] Eclipse  
<http://www.eclipse.org/>(2016.2.1 参照)
- [12] Microsoft Visual Studio  
<http://www.microsoft.com/ja-jp/>(2016.2.3 参照)
- [13] HSQLDB  
<http://hsqldb.org/>(2016.2.6 参照)