

Realization of Conversation Support System for
Social Withdrawal

メタデータ	言語: jpn 出版者: 公開日: 2021-11-04 キーワード (Ja): キーワード (En): 作成者: 池本, 武史, 黒岩, 丈介, 小高, 知宏, 諏訪, いずみ, 白井, 治彦 メールアドレス: 所属:
URL	http://hdl.handle.net/10098/00028842

ひきこもり者支援のための 会話システムの実現に向けて

池本 武史* 黒岩 丈介** 小高 知宏** 諏訪 いずみ*** 白井 治彦****

Realization of Conversation Support System for Social Withdrawal

Takeshi IKEMOTO*, Jousuke KUROIWA**, Tomohiro ODAKA**
Izumi SUWA*** and Haruhiko SHIRAI****

(Received September 30, 2021)

In this paper, we develop a conversation support system with virtual agents for socially withdrawn persons. In order to solve a problem of withdrawn persons it would be important to provide an opportunity for conversation. In this study, we develop a conversation system on Unity, which has functions to converse using chatting or listening while empathizing with the user, and to record the contents of the conversation and provide them to counselors and other supporters.

Key words : Virtual Agent , Support System , Socially Withdrawn Persons

1. はじめに

本論文では、仮想エージェントを用いた、ひきこもり者支援のための会話システムについて研究を行なった。

ひきこもりとは、厚生労働省によると、「社会参加(修学・就労・交遊など)を回避し、原則的には6ヶ月以上にわたっておおむね家庭にとどまり続ける状態」のことである。^[1] ひきこもり問題は1970年ごろに認知され始め、1990年代の終わりまで若者の問題として日本社会で広く扱われていた。だが、時が経ち、当時の若者が中高年になっていき、現在は中高年のひきこもりが多く期間が長期化している。こうしたひきこもりの長期高齢化は、「8050問題」と呼ばれる社会問題に繋がる。8050問題とは、50代のひきこもりの中高年の子供と80年代の高齢の親が孤立するという問題であ

る。2018年12月の内閣府調査によると、40歳から64歳のひきこもりの人は、全国推計で61万人になる。^[2]

ひきこもりの長期化の問題として、まず、社会的な時間感覚が失われて生活リズムがみだれる。次第に人とのコミュニケーションが不足する。そして、気力がなくなりさらに生活リズムがみだれる、という悪循環に陥ることが問題として挙げられる。この問題を解決するためのアプローチとして、我々は、ひきこもり者の会話のきっかけ作りが必要であると考えた。

先行研究で、自己表現が苦手な児童を対象とした、仮想エージェントを用いたバーチャル会話トレーニングの研究が行われている。^[4] これは、音声認識を用いて男子生徒・女子生徒・校長先生の3人を想定した仮想エージェントと会話トレーニングするシステムである。仮想エージェントとは、ユーザーの問いかけに、人工知能、機械学習、自然言語処理などを介して対応する、CGやアニメーションにより作成した仮想のキャラクターのことである。よって、我々は、ひきこもり者と対象として仮想エージェントを用いた会話システムが開発可能ではないかと考えた。また、ひきこもり者支援システムの先行事例として、「Warokuパブリックヘルス」というクラウド型相談業務システムがある。^[5] これは、支援団体がそれぞれ作成・保管し

* 大学院工学研究科 知能社会基礎工学専攻

** 知能システム工学講座

*** 仁愛女子短期大学

**** 福井大学

* Fundamental Engineering for Knowledge-Based Society,
Graduate School of Engineering,
Graduate School of Engineering

** Department of Human and Artificial Intelligent Systems

*** Jin-ai Women's College

**** University of Fukui

ていた相談記録や成育環境情報をクラウド上で保存し、支援窓口間で共有するシステムである。この事例から、ひきこもり者と会話システムのやり取りの記録を保持できる機能が重要だと考えた。将来、このような情報共有システムに会話記録を保持できれば、その情報をカウンセラーがひきこもり者との会話に生かすことができると考えるからである。

以上により、本研究の課題であるひきこもり者の会話のきっかけ作りのために、仮想エージェントを用いた、会話ログ保存が可能な会話システムの開発を行うことにした。具体的には、まずひきこもり者が、テキスト入力や音声認識を通じて仮想エージェントと会話を行い、会話のきっかけ作りをする。そして、この時の会話をログ保存して、その情報をカウンセラーがひきこもり者との会話に生かすことができるシステムを開発する。

2. システムの機能分析

本研究では、ひきこもり者の会話のきっかけ作りを行う会話システムを開発する。この章では、会話システムに必要な機能について述べる。

ひきこもり者支援のための会話システムとして求められる機能を考えるにあたって、カウンセリングの手法に注目した。カウンセリングの初期段階では、対象者の抱える問題の明確化し、問題の設定を行うことが重要である。^[3] これを行うには、ひきこもり者と信頼を築き、情報を集めることが重要であると言われている。よって、ひきこもり者の情報収集を行う機能が重要だと考える。

ひきこもり者の情報収集の方法として、カウンセリングの手法である、雑談と傾聴に注目した。雑談は、自己に関することや会話者の興味・関心など広範な話を扱うことができる。^[6] よって、雑談はひきこもり者の特徴を知るための情報収集に必要な機能だと考えた。

また傾聴とは、相手の話を否定せず、立場や気持ちに共感しながら会話することである。よって、傾聴を用いて話のペースや気持ちに合わせながら話すことで、ひきこもり者との信頼関係の構築に繋がるのではないかと考えた。

これらを踏まえて、本研究で目指す会話システムは以下のように考えた。エージェントが相手に「共感」しつつ、「雑談」と「傾聴」を用いて会話をして「情報収集」する。そして、ひきこもり者の「会話のきっかけ作り」を行う、システムである。目指すシステムのイメージ図を図1で示す。ひきこもり者と会話システムとの会話のやり取りを通じ、情報収集を行う。そして、システムから提供された情報をカウンセラーは

ひきこもり者との会話内容に利用するというシステムの流れである。以上により、本システムに必要な機能は大きく分けて以下の4つである。

1. **雑談会話機能**：ユーザーが入力した会話文に対して、雑談文を生成しエージェントが応答する機能である。
2. **傾聴会話機能**：ユーザーが入力した会話文に対して、傾聴文を生成しエージェントが応答する機能である。
3. **感情応答機能**：ユーザーが会話入力した際の感情値を抽出し、エージェントが感情値に応じた応答をする機能である。この機能によって、ユーザーに共感をしながら応答することが可能になると考える。
4. **会話ログ保存機能**：ユーザーの会話文とエージェントの応答文をログとして保存する機能である。これにより、ユーザーの興味・関心のある話題について情報収集することが可能になると考える。また、カウンセラー等に支援者に情報を提供することも可能であると考えられる。

3. ひきこもり者支援のための会話システムの設計

システムの構成とデータの流れについて述べる。2章で述べた4つの必要な機能をふまえてシステム開発するために、以下のより詳細な11の機能に分けて考えた。また、今回実装したシステムの構成図を図2で示す。

3.1 発話文入力機能

テキスト文入力でユーザーの発話文を受け取る機能である。受け取った発話文は、応答文生成機能と会話ログ保存機能に送られる。またテキスト文入力時、感情分析機能を実行させる。データの流れとしては、GUI上のテキスト入力フォーム上で入力された文字

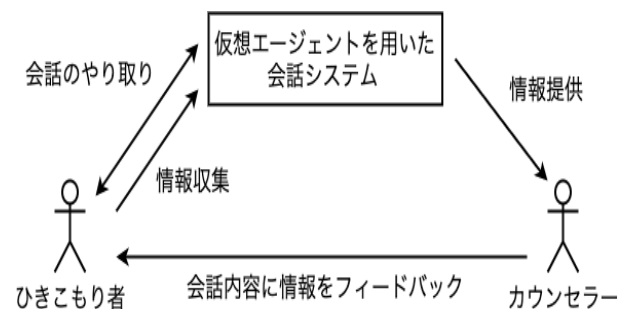


図1 会話システムのイメージ図

列データを、応答文生成機能と会話ログ保存機能に送る流れである。

3.2 音声認識機能

マイクからの音声入力によってユーザーの発話音声を文字起こしして、発話文として受け取る機能である。受け取った発話文は、応答文生成機能と会話ログ保存機能に送られる。また音声入力時、感情分析機能を実行させる。データの流れとしては、マイク入力した発話文の音声データを API に送り、API で変換された発話文の文字列データを受け取って、応答文生成機能と会話ログ保存機能に送る流れである。

3.3 感情分析機能

テキスト文入力か音声入力をした時のユーザーの表情画像を端末のカメラから取得し、感情値を分析する機能である。表情画像から抽出した感情値は、エージェント応答機能に送られる。データの流れとしては、GUI 上で取得した表情画像を API に送り、API で抽出された最も可能性の高い感情の種類情報をエージェント応答機能に送る流れである。

3.4 応答文生成機能 (雑談文生成)

発話文入力機能や音声認識機能から送られてきたユーザーの発話文に対して、エージェントの雑談応答文を生成する機能である。生成した雑談応答文は、音声合成機能と会話ログ保存機能と応答文表示機能に送られる。データの流れとしては、発話文入力機能や音声認識機能から送られてきた、発話文の文字列データを API に送り、API で生成した雑談応答文の文字列データを、音声合成機能と会話ログ保存機能と応答文表示機能に送る流れである。

3.5 応答文生成機能 (傾聴文生成)

発話文入力機能や音声認識機能から送られてきたユーザーの発話文に対して、エージェントの傾聴応答文を生成する機能である。生成した傾聴応答文は、音声合成機能と会話ログ保存機能と応答文表示機能に送られる。データの流れとしては、発話文入力機能や音声認識機能から送られてきた、発話文の文字列データを API に送り、API で生成した傾聴応答文の文字列データを、音声合成機能と会話ログ保存機能と応答文表示機能に送る流れである。

3.6 音声合成機能

応答文生成機能から受け取ったエージェントの応答文を合成音声に変換する機能である。生成した合成音声は、エージェント応答機能に送られる。データの流れとしては、応答文生成機能から受け取ったエージェントの応答文の文字列データを API に送り、API で音

声データに変換されエージェント応答機能に送る流れである。

3.7 エージェント応答機能

感情分析機能から受け取った感情値や音声合成機能から受け取った応答文の音声に応じ、エージェントに表情や身体の動作と発話を指示する機能である。データの流れとしては、感情分析機能から受け取った最も可能性の高い感情の種類情報をアニメーション制御機能に送り、それに合う表情や身体の動作のアニメーションファイルを受け取ってエージェントを制御する。また、音声合成機能から受け取った応答文の音声データをリップシンク機能に送り、エージェントの口を動かすパラメータを受け取って、エージェントに発話の指示を出すという流れである。

3.8 アニメーション制御機能

エージェント応答機能からのリクエストに応じ、エージェントの表情と体の動きを制御する機能である。データの流れとしては、エージェント応答機能から受け取った感情の種類情報に合った、表情や身体の動作のアニメーションファイルをエージェント応答機能に送信する流れである。

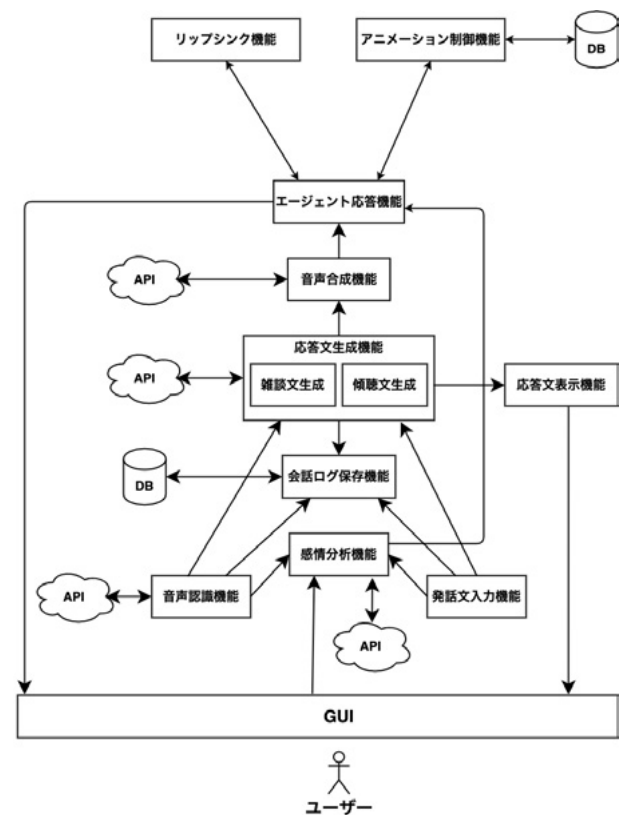


図 2 システムの構成図

3.9 リップシンク機能

エージェント応答機能から受け取った音声データに応じアバターの口の動きを制御する機能である。データの流れとしては、エージェント応答機能から受け取った音声データを再生して音域を測定し、エージェントの口を動かすパラメーターを決めてエージェント応答機能に送るという流れである。

3.10 応答文表示機能

応答文生成機能から受け取ったエージェントの応答文を画面表示する機能である。データの流れとしては、応答文生成機能から受け取ったエージェントの応答文の文字列データを、GUIのテキスト文表示パネルに送るという流れである。

3.11 会話ログ保存機能

発話文入力機能や音声認識機能から送られたユーザーの発話文と、応答文生成機能から送られたエージェントの応答文を外部に保存する機能である。データの流れとしては、ユーザーの発話文とエージェントの応答文の文字列データと送信した時間を外部ファイルに保存するという流れである。

4. ひきこもり者のための会話システムの実装

4.1 システムの開発環境

本システムの開発に用いたツールやライブラリについて述べる。開発 PC は MacBook Pro 2019 を使用する。総合開発環境者 Unity を利用し、開発エディタには、Visual Studio Code を使用する。プログラミング言語には C# と Python を用いる。

4.1.1 Unity について

Unity とは、Unity Technologies が開発・販売している、Web ブラウザ、デスクトップ PC、ゲーム機、モバイル向けのゲームやアプリケーションなど、複数の機種に対応したゲームエンジンである。ゲームエンジンとは、ゲームを簡単に作るための支援をするツールで、ゲームに必要な映像や音などの処理を行い、開発を効率化してくれる。

4.1.2 Visual Studio Code について

VSCode (Visual Studio Code) とは Microsoft が開発している、Windows・Linux・macOS 用のソースコードエディタである。このエディタは誰でも無償で利用できる。また、カスタマイズ性が高く、テーマやキーボードショートカット、環境設定を変更することができる。他にも、拡張機能をインストールして機能追加をすることができる。今回は、拡張機能によって Unity と連携し C# と Python のプログラム作成を行う。

4.1.3 使用する公開 API (Application Programming Interface) について

本システムで利用する API について述べる。API とは、ソフトウェアやプログラム、Web サービスの間をつなぐインターフェースのことである。つまり API を用いることで例えば、異なるソフトウェアやサービス間でチャット機能を共有したり、片方から数値データを取り込んで別のプログラムでそれを解析したりできるようになる。API を公開するとは、ソフトウェアに API という窓口を作り、外部のアプリと連携できる状態にするということである。本システムの開発では以下の、4 つの 公開 API を使用する。

- **Google Cloud Speech-to-Text** : Google 社が提供する、音声文字変換 API である。音声データをこの API に送信すると、テキスト文に変換されて受け取ることができる。この API を用いることで音声認識による発話文生成・入力が可能になる。
- **A3RT(読みは「アート」)** : 機械学習や深層学習といった技術を、リクルートテクノロジーズ社が外部提供している API 群である。現在、11 種類の API が提供されている。その中でも本研究では、Talk API という API を使用する。この API は、チャットボットを作成するための API である。テキスト文をこの API に送信すると、RNN (Recurrent Neural Network) を用いて応答文を生成して受け取ることができる。RNN とは、数値の時系列データなどを認識するように設計された、ニューラルネットワークモデルである。つまり RNN は、過去の情報を記憶しその情報に従い新しい事象を処理することが可能である。この API を用いることで雑談文生成が可能になる。
- **Amazon Polly** : Amazon Web Services 社が提供する、深層学習を用いて文章をリアルな音声に変換するサービスである。この API では、数十におよぶ言語と自然な男女の音声を利用することができる。テキスト文をこの API に送信すると音声データに変換され、受け取ることができる。
- **Amazon Rekognition** : Amazon Web Services 社が提供する、機械学習を使用して画像と動画の分析を自動化するサービスである。画像の検索・検証・整理を実施し、画像内の物体・場面・顔の検出、有名人の認識、画像の比較などを行うことができる。例えば、表情画像をこの API に送信すると、画像から感情分析されて感情値を受け取ることができる。また感情値は次の 8 種類に分類さ

れる。Happy (幸せ)・Angry (怒り)・Surprised (驚き)・Confused (困惑)・Disgusted (呆れる)・Calm (穏やか)・Fear (恐れ)・Sad (悲しい)の8種類である。

4.1.4 Flask について

他にも、Unity のプロジェクトフォルダ外の Python ファイルを Unity で実行するために、Flask を用いた。Flask とは、Python で利用できる Web アプリケーションフレームワークである。これによって Python で簡単な Web API を作成することができる。そして、Unity で実行する C# のプログラムから API 通信を行うことで作成した Web API を利用することが可能になる。API の中でも、HTTP 通信によりやりとりを行うものを Web API という。

4.1.5 MeCab について

また、形態素解析エンジンである MeCab も利用する。MeCab とは、京都大学情報学研究科と日本電信電話株式会社コミュニケーション科学基礎研究所の共同研究で開発されたツールである。形態素解析とは、言語の文法や品詞の情報を利用し、文章を形態素(意味を持つ最小単位の単語)に分解する解析である。例えば、「趣味の話をしたと思います。」という文章を、MacBook のターミナル上で MeCab を実行し解析した結果は、「趣味 | の | 話 | を | し | たい | と | 思 | い | ます | 。」と、品詞ごとに 10 個の形態素に分けられる。

4.2 Unity の導入方法

以下で、Unity の導入手順について述べる。

1. ダウンロードページから、Unity Hub をダウンロードする。Unity Hub とは、Unity のバージョン管理ツールであり、Unity をアップデートしたら動かなくなるという事態を回避するために必要である。
2. Unity Hub をダウンロードできたら、それを起動し、メニューのインストールから、Unity をインストールする。今回は、Unity 2020.3.6.f1 バージョンをインストールする。
3. インストールが終わった後、メニューの [プロジェクト] → [新規作成] の手順で、プロジェクト新規作成画面に移動する。プロジェクトとは、Unity で一つのゲームを製作するのに、使ったもの全てを収納したフォルダのことである。
4. プロジェクト新規作成画面で、プロジェクトの名前とプロジェクトの保存先を指定する。プロジェクトのテンプレートは、本研究では 3D を選択する。
5. 作成を選ぶことで、新規のプロジェクトが作成され、Unity を使用できるようになる。

4.3 Visual Studio Code の環境構築

VSCoDe の環境構築手順について述べる。

1. **VSCoDe をインストールする**：VSCoDe の公式ホームページから、ソフトウェアをインストールする。Windows・Linux・macOS の三つの OS がダウンロード可能なので、開発に利用する端末に合わせてダウンロードする。
2. **スクリプトを開くエディタを VSCoDe に変更**：Unity のプロジェクトを開きメニュー欄で「Unity」→「preferences」→「External Tools」→「External Script Editor」を選択しメニューを開く。メニューの中から VSCoDe に変更する。これによって、Unity の右クリックメニュー「Open C# Project」を押したときや、Unity 上で C# ファイルをダブルクリックしたとき、VSCoDe が開くようになる。
3. **C# の拡張機能を導入**：VSCoDe を起動し、画面左のメニューから「拡張機能」を選択して検索候補に「C#」と入力し、C# の拡張機能をインストールする。これによって、VSCoDe 上で C のスクリプトのデバッグと、コード補完が可能になる。
4. **Debugger for Unity の拡張機能を導入**：C# の拡張機能を導入した時と同じ手順で、Debugger for Unity の拡張機能をインストールする。これによって、Visual Studio Code で Unity をデバッグできるようになる。
5. **Python の拡張機能を導入**：C# の拡張機能を導入した時と同じ手順で、Python の拡張機能をインストールする。これによって、VSCoDe 上で Python のスクリプトのデバックと、コード補完が可能になる。
6. **Flask でローカルサーバー立ち上げ**：使用 PC のターミナルを開き、Flask をインストールする。これによって PC にローカルサーバーを立ち上げ、Unity から Python のスクリプトに API 通信できるようになる。

7. **MeCab を導入**：まず、使用 PC のターミナルを開き、MeCab をインストールする。その後、MeCab で利用する中で最もメジャーな辞書の一つである IPA 辞書を使用可能にするため、mecab-ipadic をインストールする。そして、MeCab を Python から利用できるようにするため、mecab-python3 をインストールする。最後に、MeCab の性能を実用レベルまで引き上げるために、neologd 辞書をインストールする。これによって、VSCode 上で MeCab を利用することができるようになる。

4.4 Unity ちゃんの利用

本システムでは、仮想エージェントに Unity Technologies Japan 社が提供するオリジナルキャラクター、「Unity ちゃん」を使用する。このキャラクターは身体パーツの 3D モデルデータが詳細に作成されており、ゲーム上で柔軟な操作がしやすいモデルである。ダウンロード後、Unity の Project ウィンドウにドラッグアンドドロップし Unity にインポートすることで、Unity ちゃんを使用することができるようになる。

4.5 各機能の実装

3 章で述べた 11 の機能を Unity で実装する方法について以下で述べる。

4.5.1 発話入力機能の実装

まず Unity の Project ウィンドウ上に、発話文の入力フォームと発話文送信用ボタンを配置する。次に、以下のことを行う C# のスクリプトを作成する。まず、発話文送信ボタンを押した際にフォームに入力した文字列データを、応答文生成機能と会話ログ保存機能を実行する関数に渡し実行させる。また、発話文送信ボタンを押した際に感情分析機能を実行する関数を実行させる。

4.5.2 音声認識機能の実装

まず Unity の Project ウィンドウ上に、音声入力用ボタンを配置する。次に、以下のことを行う C# のスクリプトを作成する。まず、音声入力用ボタンを押している間はマイク入力を起動させる。次にボタンを離れた時、それまで入力した音声データを音声文字変換 API の Google Cloud Speech-to-Text に送信する。またボタンを離れた時、感情分析機能を実行する関数を実行させる。そして、API から変換した文字列データを受け取る。最後にそのデータを、応答文生成機能と会話ログ保存機能を実行する関数に渡し実行させる。

4.5.3 感情分析機能の実装

まず、Unity の Project ウィンドウ上に、PC の内部カメラで撮影した表情画像を表示するパネルを配置する。次に、以下のことを行う C# のスクリプトを作成する。まず、発話文送信用ボタンや音声入力用ボタンを押した時にパネルに表示される表情画像を取得する。次に、取得した表情画像を感情分析 API である Amazon Rekognition に送信する。最後に API から受け取った 8 種類の感情値の中で、最も値の大きい感情の種類情報をエージェント応答機能に送信する。

4.5.4 応答文生成機能(雑談文生成)の実装

まず、Unity の Project ウィンドウ上に、以下の二つのトグルボタンを配置する。1 つ目は、テキスト文入力での雑談文生成を行う機能を実行可能にするボタン。2 つ目は、音声入力での雑談文生成を行う機能を実行可能にするボタンである。次に以下のことを行う C# のスクリプトを作成する。まず、発話文入力機能や音声認識機能から送られてきた、発話文の文字列データを、雑談文生成 API である A3RT に送信する。そして、API から応答文の文字列データを受け取る。最後に、そのデータを音声合成機能と応答文表示機能と会話ログ保存機能を実行するそれぞれの関数に送信する。

4.5.5 応答文生成機能(傾聴文生成)の実装

まず、Unity の Project ウィンドウ上に、以下のトグルボタンを配置する。テキスト文入力での傾聴文生成を行う機能を実行可能にするボタンである。上記で述べた 2 つのトグルボタンを合わせて、3 つのトグルボタンを切り替えることで、実行する機能を切り替え可能にする。

次に以下のことを行う C# のスクリプトを作成する。まず、発話文入力機能から送られてきた、発話文の文字列データを、Flask を介して Python のスクリプトに送信する。次に、Python のスクリプトによって生成された傾聴応答文の文字列データを Flask を介して受け取る。最後に、そのデータを音声合成機能と応答文表示機能と会話ログ保存機能を実行するそれぞれの関数に送信する。

また、以下のことを行う Python のスクリプトを作成する。まず、Flask を介して受け取った発話文の文字列データを、MeCab によって形態素に分解する。次に、事前に作成した変換辞書に当てはまる、特定の形態素を変換する。例えば、「趣味 | の | 話 | を | し | たい | と | 思い | ます | 。」と 10 個に分割された形態素を、「趣味 | の | 話 | を | し | たい | と | 思う | の | ですね | 。」と変換する。このように、相手の話の語尾を書き換え

て繰り返すような傾聴応答文を生成する。そして最後に、生成した傾聴応答文を文字列データとして、Flask を介して C# のスクリプトに送信する。

4.5.6 音声合成機能の実装

以下のことを行う C# のスクリプトを作成する。まず、応答文生成機能から受け取ったエージェントの応答文を、文字列データとして音声合成 API の Amazon Polly に送信する。次に、API から音声データを受け取る。最後に、その音声データをエージェント応答機能を実行する関数に送信する。

4.5.7 エージェント応答機能の実装

以下のことを行う C# のスクリプトを作成する。まず、音声合成機能から受け取った応答文の音声データをリップシンク機能を実行する関数に送信する。次に、感情分析機能から送られてきた、最も感情値が高い感情の種類の情報、アニメーション制御機能を実行する関数に送信する。

4.5.8 アニメーション制御機能の実装

Unity のアニメーション管理機能である Animator Controller を利用する。まず、Animator Controller を開き、エージェントの胴体を動かすアニメーション管理をする、「Base Layer」と、表情のアニメーションを管理する、「Face Layer」を追加する。次に、それぞれの Layer に 8 種類のアニメーションファイルを設置する。そして、以下のことを行う C# のスクリプトを作成する。エージェント応答機能から受け取った感情の情報に応じたアニメーションファイルを実行し、エージェントの動作を変化させる。

4.5.9 リップシンク機能の実装

以下のことを行う C# のスクリプトを作成する。エージェント応答機能から受け取った応答文の音声データを再生し、音域に応じて、エージェントの口を開くパラメーターを変化させる。

4.5.10 応答文表示機能の実装

まず、Unity の Project ウィンドウ上に、応答文を表示させるパネルを配置する。次に以下のことを行う C# のスクリプトを作成する。発話文入力機能や音声認識機能から送られてきたユーザーの発話文の文字列データをパネルに表示させる。

4.5.11 会話ログ保存機能の実装

以下のことを行う C# のスクリプトを作成する。発話文入力機能や音声認識機能から送られたエージェントの発話文と、応答文生成機能から送られたエージェントの応答文を、CSV ファイルに文字列データとして送信し保存する。

5. ひきこもり者支援のためのシステムの動作実験

5.1 動作実験の概要

今回、Unity にて開発した会話システムが正常に起動するかどうかノート PC にて確認する。動作実験に使用する PC は MacBook Pro 2019 である。前章で実装した機能群が実際に意図通り動作するのを確認するのが本実験の目的である。

5.2 動作実験の結果

動作実験の結果、三つの結果を得ることができた。まず一つ目は、エージェントによる雑談と傾聴応答機能によりユーザーと簡単な会話が可能になったことである。続いて二つ目は、ユーザーの表情画像から感情分析し、それに合わせたエージェントの応答が可能になったことである。最後に三つ目は、会話ログを CSV 保存することにより、ユーザーの情報収集が可能になったことである。



図 3 雑談会話機能の実行例 1



図 4 雑談会話機能の実行例 2

エージェントの雑談会話機能の実行例を図 3 と図 4 で、会話ログ保存機能の実行例を図 5 で示す。図 3 で

134	クライアント	アバター	2021/7/13
135	こんにちは		10:20:14 AM
136		こんにちは	10:20:15 AM
137	今日は何をしますか		10:20:46 AM
138		まだ決めていません	10:20:47 AM
139	疲れました		10:21:47 AM
140		お疲れ様です	10:21:47 AM

図5 会話ログ保存機能の実行例

は、「テキストで雑談」という項目にチェックを入れ、「今日は何をしますか?」とテキスト入力し、「会話する」ボタンを押して送信を行なっている。またこの時、PCの内部カメラから取得した表情画像を感情分析APIに送信を行なっている。

その後、図4では、「今日は何をしますか?」という発話文に対して「掃除します」という応答文がAPIで生成され、画面に表示されている。またこの時、エージェントが応答文の発話を行っている。それに加えて、表情画像からAPIによって感情の情報を抽出し、それに応じてエージェントが表情と身体の動きを変化させている。図5では、CSVファイルにて、ユーザーの入力文とエージェントの応答文と送信された時間が記録されている。項目はクライアント・アバター・今日の日付の3つに分けられている。

6. 考察

動作実験結果を踏まえて今後の課題として、一つ目の結果に関して、今回のシステム開発では、会話の中で雑談と傾聴を用いるタイミングについて分からなかったため、各機能の確認で雑談文生成と傾聴文生成の機能を分けた。よって、そのタイミングについて今後考える必要があることが分かった。また、今回は辞書マッチによる変換法のみで傾聴文生成を行なった。よって、品詞変換による生成法も用いることで、傾聴文生成パターンをより増やすことができると考える。続いて二つ目の結果に関して、感情分類結果に応じて、応答文の語尾を変える機能の実装が今後必要だと考える。これによって、現時点ではエージェントの表情と動作の変化のみであるが、発話内容も感情に応じて変えることができるからである。最後に三つ目の結果に関して、会話ログを見てエージェントの応答をコントロール可能にするような機能が必要だと考える。これによって、ひきこもり者の状態に合わせて、カウンセラー等に支援者に情報を提供しやすくなるのと考えられるからである。

7. まとめと今後の課題

今回、仮想エージェントを用いたひきこもり者支援のための会話システムをUnityで開発した。現在、

1. テキスト打ち込みや音声認識によって入力した会話文から、雑談文や傾聴文を生成し、エージェントと短文で会話すること。
2. 表情画像から抽出した感情に合わせてエージェントの表情や動作の変化をすること。
3. 会話ログを外部ファイルに保存すること。

が可能となっている。今後は、会話の中で雑談と傾聴を用いるタイミングについて考えることや、感情分類結果に応じ、応答文の語尾を変える機能が必要だと分かった。他にも、会話ログを見てエージェントの応答をコントロール可能にするような機能が必要だと考える。

参考文献

- [1] ひきこもりの評価・支援に関するガイドライン、<https://www.mhlw.go.jp/content/12000000/000807675.pdf> (2021/09/08)
- [2] Dinh Quynh Trang, Nguyen Phuong Thuy: 現代日本における8050問題 - 中高年ひきこもりへの支援活動に着目して-, ベトナム国家大学ハノイ校日越大学 地域研究プログラム日本研究専攻 修士論文, (2020)
- [3] 江刺利彦, 久保村千明, 服部峻, 亀田弘之: 擬人化エージェントによるマルチモーダルなカウンセリングシステムの構築, 情報処理学会創立50周年記念(第72回)全国大会, pp.623-624, (2010)
- [4] 伊藤基晴, 伊藤大河, 山本利一: 小学校通級指導教室に通う児童の会話力を育てる教育実践—音声認識と3Dキャラクターを用いたバーチャル会話トレーニング—, 埼玉大学教育学部附属教育実践総合センター紀要16, pp.9-15, (2017)
- [5] Waroku パブリックヘルス | 株式会社レスコ, https://rescho.co.jp/product/waroku/public_health.html (2021/09/08)
- [6] 星野春香, 松本知香: 心理臨床面接における雑談の可能性についての一考察, 京都大学大学院教育学研究科附属臨床教育実践研究センター紀要24, pp.110-117, (2021)