

NH-chat 導入手順書

作成者：奥野 尚己

2021 年 5 月 17 日

1 NH-chat の概要

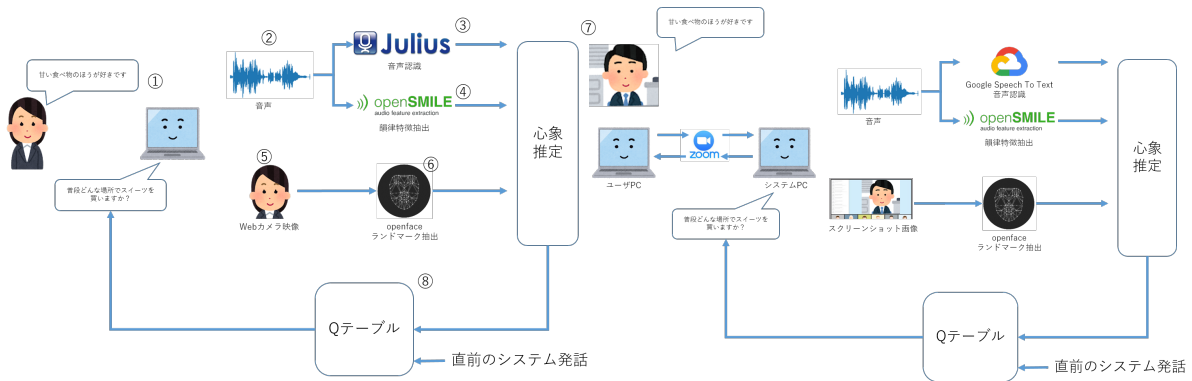


図 1 従来 NH-chat

図 2 遠隔 NH-chat

NH-chat とは MMDAgent を用いた音声対話システムである。従来の NH-chat の構成を図 1 に示す。NH-chat の対話におけるシステム発話選択には言語情報、韻律情報、顔画像情報が用いられている。これらの情報を用いて推定されたユーザ心象と過去のシステム発話などを学習済み Q テーブルに入力し、次のシステム発話を選択している。従来の NH-chat はシステムの動作する PC と対面した状態で使用されることを前提としている。ユーザの顔画像は Web カメラの映像を取得することでシステムに入力され、システム発話選択に用いる各情報の取得には以下のアプリが使用されている。

- 言語情報 (音声認識)：Julius
- 韻律情報：openSMILE
- 顔画像情報：OpenFace

遠隔 NH-chat は遠隔かつ自動で対話データを収集することを目的に作成したシステムである。遠隔 NH-chat の構成を図 2 に示す。従来のシステムとの違いは、音声認識に Google Speech To Text を用いる点と、被験者が遠隔で参加する点である。顔画像情報の取得にはスクリーンショッ

トを用いている．具体的には，Zoom で被験者が映るウィンドウのスクリーンショットを取得し，その画像を OpneFace に逐次的に入力することで顔画像情報を得ている．

本テキストで導入するシステムは，従来の用途（対面して web カメラを使用する方法）でも遠隔でも使用できる．具体的な使用方法については実験の手順書 (明日書きます) を参照．

2 動作環境

windows 8.1, windows 10 で動作確認を行った．テスト環境では Python3.6 を使用している．テスト環境で使用したモジュール等についての詳細は以下の手順 1 でクローンしたリポジトリ内 NH-chat_env.yml を参照．

3 導入手順

NH-chat を導入する手順を示す．導入の大まかな手順は以下の通りとなる．

1. git hub からリポジトリをクローン
プログラムファイルや学習済みモデルなどをダウンロードする
2. conda を用いた仮想環境の構築
開発者のテスト環境と同等の環境を構築する．
3. 必要なアプリ等のダウンロード
4. モジュールの導入，動作確認

各工程の詳細を以下に示す．

3.1 git hub からリポジトリをクローン

以下のコマンドを実行し git hub からリポジトリをクローンする．

```
git clone https://github.com/elprimo041/NH-chat.git
```

フォルダ構造および各ファイル，フォルダの機能は README(明日書きます) 参照．

3.2 conda を用いた仮想環境の構築

開発者環境を再現するために conda を用いる．conda を用いて仮想環境を構築すると，開発環境と同じバージョンの Python，モジュールを一括でインストールすることができる．まず miniconda をダウンロードし，次にクローンしたリポジトリ内にある環境情報が記載されたファイルを使用して開発環境と同等の仮想環境を構築する．

3.2.1 miniconda のインストール

すでに Anaconda や miniconda がインストール済みで conda コマンドが使用できる場合はこの工程をスキップする．

以下のリンクから miniconda をインストールする。

<https://docs.conda.io/en/latest/miniconda.html>

miniconda インストール時の Python のバージョンは任意のものでよい。ここで選択した Python のバージョンはシステム実行時に使用されるものとは異なるためである。次の工程で開発者と同じバージョン (Python3.6.10) が自動的にダウンロードされる。

3.2.2 仮想環境の構築

Anaconda Prompt でクローンした NH-chat のフォルダに移動する。(NH-chat.env.yml のあるフォルダ) 以下のコマンドを実行し仮想環境を構築する。

```
conda env create -n NH-chat -f NH-chat.env.yml
```

システムの実行に必要な Python とモジュールが一括インストールされる。

3.3 必要なアプリ等のダウンロード

NH-chat の動作に必要なアプリのダウンロードやパスの設定等を行う。まず、Google 音声認識以外のアプリで共通のパスの設定方法について説明する。次に各アプリの具体的な導入方法を示す。導入するアプリは OpenFace, openSMILE, MMDAgen, Google Cloud SDK の 4 つである。Julius 音声認識を使用する場合は dictation-kit の導入が追加が必要となる。

3.3.1 パスの設定方法

	A	B	C
1	MMDAgent	C:/Users/okuno/Desktop/MMDchat/MMDAgent_win32-1.4/MMDAgent_win32-1.4	MMDAgent.exeがあるフォルダ
2	openface		FaceLandmarkImg.exeがあるフォルダ
3	opensmile	C:/Users/okuno/Desktop/MMDchat/opensmile-3.0-win-x64/opensmile-3.0-win-x64	bin,configフォルダのあるフォルダ
4	dictation	C:/Users/kaldi/Desktop/okuno%dictation_kit_okuno	bin,configフォルダのあるフォルダ

図 3 パスの設定方法

以降の工程で行うパスの設定方法について説明する。本システムではアプリの絶対パスをクロー

ンしたリポジトリ内にある refData/path.csv に記載する。OpenFace のパスを設定する場合の操作を図 3 に示す。OpenFace の場合、ダウンロード、展開後に FaceLandmarkImg.exe のあるフォルダの絶対パスをコピーし、csv ファイルの該当する箇所にコピーする。他のアプリでも同様の方法でパスを設定する。

3.3.2 OpenFace のダウンロード

以下のリンクから OpenFace をダウンロードし任意のフォルダに展開する。

https://github.com/TadasBaltrusaitis/OpenFace/releases/download/OpenFace_2.2.0/OpenFace_2.2.0_win_x64.zip

FaceLandmarkImg.exe があるフォルダのパスを、クローンした NH-chat 下の refData/path.csv の該当する箇所にコピーする。

ダウンロードした OpenFace には顔画像の推定に用いるモデルが付属していない。モデルをダウンロードするために download_model.ps1 を右クリックし Power Shell で実行する。

3.3.3 openSMILE のダウンロード

以下のリンクから opensmile をダウンロードし任意のフォルダに展開する。

<https://github.com/audeering/opensmile/releases/download/v3.0.0/opensmile-3.0-win-x64.zip>

bin, config フォルダのあるフォルダのパスをクローンした NH-chat 下の refData/path.csv の該当する箇所にコピーする。

3.3.4 MMDAgent のダウンロード

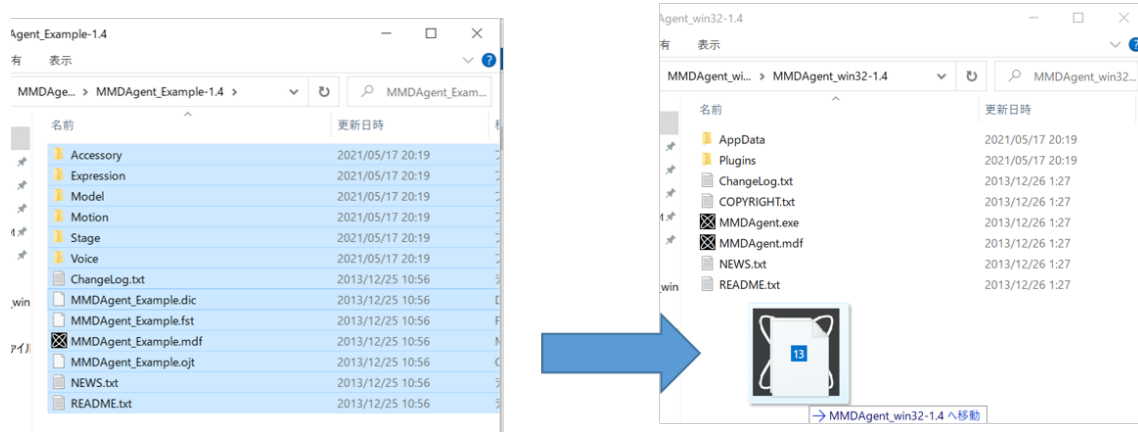


図 4 サンプルプログラムのコピーの操作

MMDAgent の本体、サンプルプログラム、ソケット通信プラグインから MMDAgent が使用できる環境を構築する。まずこれらのダウンロードの手順を示し、次に MMDAgent を使用するために行うファイルのコピーの手順を示す。

必要なファイルを以下の手順でダウンロードする。まず下記のリンクから MMDAgent の本体をダウンロードする。

https://ja.osdn.net/projects/sfnet_mmdagent/downloads/MMDAgent/MMDAgent-1.4/MMDAgent_win32-1.4.zip/

次に下記のリンクから MMDAgent のサンプルプログラムをダウンロードする。

https://ja.osdn.net/projects/sfnet_mmdagent/downloads/MMDAgent_Example/MMDAgent_Example-1.4/MMDAgent_Example-1.4.zip/

本体の MMDAgent.exe があるフォルダのパスをクローンした NH-chat 下の refData/path.csv の該当する箇所にコピーする。最後にソケット通信プラグインをダウンロードする。MMDAgent にはデフォルトではソケット通信を行う機能はないので、プラグインをダウンロードする必要がある。MMDAgent のソケット通信プラグインを以下のページからダウンロードする。

https://ux.getuploader.com/cube370/download/15/Plugin_SocketRemote.zip

それぞれのファイルを任意のフォルダに展開する。

次に MMDAgent を使用するために必要なファイルのコピーの手順を示す。NH-chat で MMDAgent を使用するには、まずサンプルプログラム内にあるファイルを MMDAgent 本体 (MMDAgent.exe) のあるフォルダにコピーする必要がある。展開した MMDAgent のサンプルプログラムに含まれるすべてのファイルを MMDAgent.exe のあるフォルダにコピーする。操作の様子を図 4 に示す。同じファイルの置き換えはスキップを選択する。

次に、ソケット通信を行えるようにするために、MMDAgent 本体の Plugins フォルダに Plugin_SocketRemote.dll をコピーする。

4 Google 音声認識のセットアップ

Google 音声認識を使用するには GCP に登録し API キーを含む json ファイルを取得する必要がある。奥野と同じ json ファイルを使用する場合、以下の 1, 2, 3 の操作は不要。4 の操作はどちらの場合でも必要となる。奥野とは別のアカウントで Google 音声認識を使用する場合は GCP(Google Cloud Platform) への登録が追加で必要となる。新たなアカウントで json ファイルを取得する手順は以下の通り。

1. Google アカウントを作成し、GCP コンソールにログインしてプロジェクトを作成する
2. Google Speech-to-Text API を有効にする
3. 「認証情報」を選択し、「認証情報を作成」から「サービスアカウントキー」をクリックして、json ファイルを作成し、任意の場所に保存する。この json ファイルを読みこむことによって API サービスを使うことができる。
4. Google Cloud SDK をインストールする。インストールの手順は以下のリンクを参照。

<https://cloud.google.com/sdk/docs/install?hl=JA#windows>

取得した json ファイルのパスを環境変数に設定する。アプリの導入で用いた csv ファイルに記載する方法とは異なるので注意。取得した json ファイルの絶対パスをコピーし、コマンドプロンプトで下記のコマンドを実行する。

SETX GOOGLE_APPLICATION_CREDENTIALS "path.to.json"

4.0.1 dictation-kit のダウンロード (任意)

Julius 音声認識を利用しない場合はスキップ.

以下のリンクから dictation-kit-4.4 をダウンロードし任意のフォルダに展開する.

<https://osdn.net/projects/julius/releases/66544>

(2020/05/14 現在バージョン 4.5 が最新ですが動作確認ができていないのは 4.4 のみです.) bin, config フォルダのあるフォルダのパスをクローンした NH-chat 下の refData/path.csv の該当する箇所にコピーする.

5 モジュールの導入, 動作確認

Anaconda Prompt を起動する. クローンしたりポジトリに移動し, 下記のコマンドで仮想環境を起動する.

```
conda activate NH-chat
```

Conda 環境が有効になると, コマンドプロンプトに (環境名) が表示される.

intro_confirm.py を実行すると環境構築が正しくできているか確かめられる. MMDAgent は初回起動時, 実行許可が必要