

# 音声認識を用いた天気予報アナウンスシステム

宗像北斗

2020 年 5 月 14 日

## 概要

本システムは、Julius によって音声認識を行い、ユーザーの希望に沿った天気情報を Web から取得し、ターミナル上で表示、あるいは音声合成にて合成された音声にてアナウンスするシステムである。

## 1 前提事項

### 1.1 使用ソフトおよび実行環境

python3 にて作成したプログラム以外に、以下のソフトおよびデータを用いた。

- Julius (ver4.5)  
音声認識ソフト ディクテーションキット (ver4.5) を使用
- Open JTalk (ver1.10)  
音声合成ソフト 音声モデルとして MMDAgent (ver1.7) を使用
- SoX (ver14.4.2)  
音声処理 (再生) ソフト

音声認識は、Julius をモジュールモードで起動し、Port:10500 にて TCP によってプログラムへと認識結果を送信することで実装している。天気予報データは Livedoor 天気情報の WebAPI に HTTP 通信を行い.json 形式でデータを取得している。また、以下の環境で動作確認済みである。

- OS: Windows 10
- ターミナル: Cygwin

### 1.2 入出力

- 入力:マイクでの音声入力
- 出力:ターミナルへの表示に加え、オプションにてスピーカーなどのオーディオデバイスからの再生。

## 2 実行方法

### 2.1 初期設定

weather.cfg に Julius で使用するコンフィグファイルおよびディクテーションキット内の音響モデルのディレクトリ、Open JTalk の辞書データおよび音声モデルのディレクトリならびにオプションを記述する。

### 2.2 実行方法

\$ python weather.py [-オプション]  
と実行するとシステムが起動する。この際同時に、バックグラウンドで Julius が起動している。以後、システムの指示に従って音声入力を加えていくことでシステムを動作させる。オプションについては、00README.txt を参照。

## 3 プログラムの仕様

### 3.1 ファイル構成

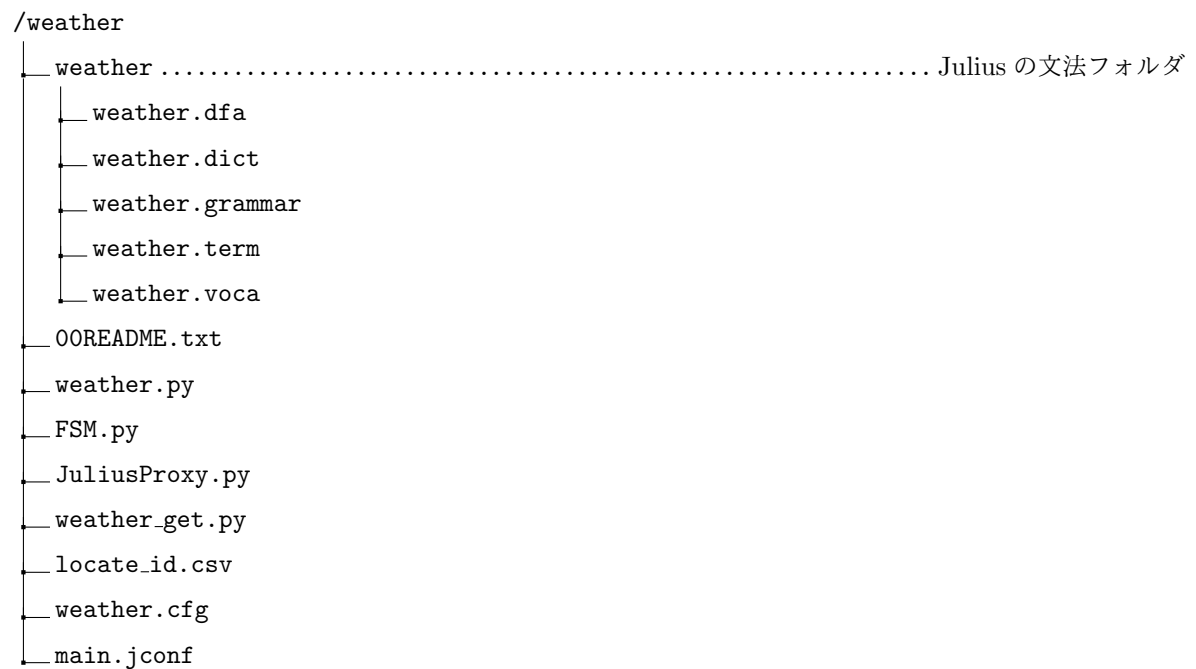


図1 ディレクトリ図

## 3.2 システム構成

- weather.py  
実行ファイル。FSM.py から提供されたオートマトンに遷移関数などを与え、JuliusProxy.py から提供された認識結果を基にオートマトンを動作させている。
- FSM.py  
オートマトンを定義し、weather.py に提供するモジュール
- JuliusProxy.py  
Julius から TCP を介して音声認識結果を受け取り、weather.py に提供するモジュール
- weather\_get.py  
weather.py のオートマトンが遷移したときに実行されるモジュール。
- TTS.py  
Open JTalk による wav ファイル生成および SoX での wav ファイルを再生するモジュール。
- locate\_id.csv  
Livedoor 天気情報にアクセスする際の位置データをまとめたファイル
- weather.cfg  
Julius, Open Jtalk 関連ファイルのディレクトリをまとめたファイル
- main.jconf  
Julius の起動オプションファイル

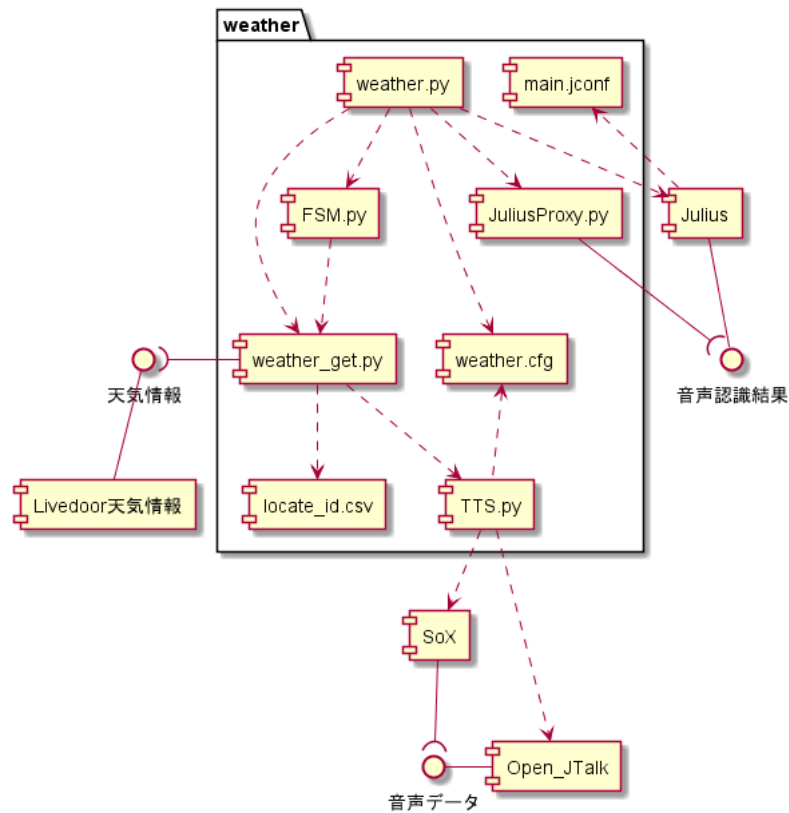


図2 コンポーネント図

### 3.3 音声認識部

辞書数は 65 個、文法数は 13 個である。辞書データには都道府県名および動詞の命令系などが含まれている。都道府県名の認識は、『○○県（都、府）』と『○○』のどちらでも認識が可能となっている。辞書データにない単語は認識できない。

### 3.4 システムの状態

システムは greet 状態で起動し、exit 状態に到達すると、動作を終了する。音声認識による命令を受け付けると、命令および現在の状態に応じた動作を行い、状態を遷移する。

- greet

システムの起動時の状態であり、以下のような命令を受け付ける。

- 『(都道府県名) の天気を教えて』

- : 指定された都道府県の天気情報を取得、表示し、その情報を保持して location 状態へ遷移する。

- 『ヘルプ』

- : ヘルプを表示し、need.help 状態へ遷移する。

- 『もういいよ』、『ありがとう』  
：終了時のあいさつをし、exit 状態へ遷移する。

- location

与えられた都道府県の天気情報を保持した状態であり、以下のような命令を受け付ける。

- 『(都道府県名) の天気を教えて』  
：再度指定された都道府県の天気情報を取得、表示し、その情報を保持して内部遷移する。
- 『明日の天気を教えて』、『あさっての天気教えて』  
：現在保持する天気情報の内、明日および明後日の天気情報を表示し、detail 状態に遷移する。
- 『詳しく教えて』、『詳細を教えて』  
：現在保持する天気情報の内、詳細についてを表示し、detail 状態に遷移する。
- 『ヘルプ』  
：ヘルプを表示し、need\_help 状態へ遷移する。
- 『もういいよ』、『ありがとう』  
：終了時のあいさつをし、exit 状態へ遷移する。

- detail

与えられた都道府県の天気情報を保持した状態であり、以下のような命令を受け付ける。

- 『(都道府県名) の天気を教えて』  
：指定された都道府県の天気情報を取得、表示し、その情報を保持して location 状態へ遷移する。
- 『明日の天気を教えて』、『あさっての天気教えて』  
：現在保持する天気情報の内、明日および明後日の天気情報を表示し、内部遷移する。
- 『詳しく教えて』、『詳細を教えて』  
：現在保持する天気情報の内、詳細についてを表示し、内部遷移する。
- 『ヘルプ』  
：ヘルプを表示し、need\_help へ遷移する。
- 『もういいよ』、『ありがとう』  
：終了時のあいさつをし、exit 状態へ遷移する。

- need\_help

- 『(都道府県名) の天気を教えて』  
：指定された都道府県の天気情報を取得、表示し、その情報を保持して location 状態へ遷移する。
- 『詳しく教えて』、『詳細を教えて』  
：現在保持する天気情報の内、詳細についてを表示し、detail 状態に遷移する。
- 『ヘルプ』  
：ヘルプを表示し、need\_help 状態へ遷移する。
- 『もういいよ』、『ありがとう』  
：終了時のあいさつをし、exit 状態へ遷移する。

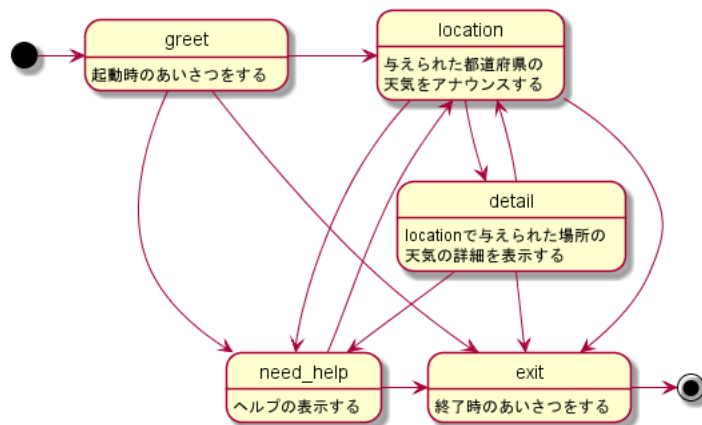


図3 状態遷移図

### 3.5 対話例

システム >> 遅くまでお疲れ様です。どこの天気を知りたいですか？

システム >> もし使い方がわからなかったら、「ヘルプ」と話してください。

話者 >> ヘルプ

システム >> まずはじめに、「(都道府県名)の天気を教えて」と話してください。するとその県の天気をお教えいたします。

システム >> 次に、「(今日、明日、あさってのいずれか)の天気を教えて」と話してください。するとその日の天気をお教えいたします。

システム >> また、「詳細を教えて」と話すと、その地域の詳しい天気をお教えいたします。

システム >> 最後に、「ありがとう」あるいは「もういいよ」と言われると、プログラムを終了します。

システム >> では知りたい場所の天気を教えてください。

話者 >> 大阪の天気を教えて

システム >> 大阪府大阪の天気です。

システム >> 本日、5月14日の天気は

システム >> 晴れで、

システム >> 最低気温は14℃で、

システム >> 最高気温は26℃です。

話者 >> 明日の天気を教えて

システム >> 明日、5月15日の天気は

システム >> 曇りで、

システム >> 最低気温はデータなしで、

システム >> 最高気温はデータなしです。

話者 >> 詳しく教えて

システム >> 詳細についてです。

システム >> 大阪府は、高気圧に覆われて晴れています。

システム >> 13 日の大阪府は、高気圧に覆われて晴れるでしょう。

システム >> 14 日の大阪府は、高気圧に覆われて晴れますが、気圧の谷の影響で夜は曇る見込みです。

システム >> 【近畿地方】

システム >> 近畿地方は、高気圧に覆われておおむね晴れています。

システム >> 13 日の近畿地方は、高気圧に覆われて晴れるでしょう。

システム >> 14 日の近畿地方は、高気圧に覆われて晴れますが、気圧の谷の影響で夜は曇る見込みです。

話者 >> もういいよ

システム >> プログラムを終了します。おやすみなさい。

## 4 工夫点、改善点など

- Julius の起動、終了を main 部分の weather.py で管理するようにしたことで、システムの起動をスムーズにすることができた。
- 各種連携ソフトのオプションを weather.cfg にて一元的に管理することで、システムの扱いやすさを向上させた。
- コマンドライン引数でのオプションを実装した。
- 音声出力時、途中で割り込んで命令を与えられるようにしたかったが、エコーキャンセルや非同期処理の難しさから断念した。
- Open JTalk で生成した wav ファイルを直接 SoX に与えることで中間生成ファイルを残さないようにしたかったが、エラーが多発したので断念せざるを得なかった。