

# 人工無脳によるチャットボットの制作

内山 柊翔 岡田 遙澄  
戸坂 隼梧 藤井 玲衣

## 1. 研究概要

コンピュータとのリアルな会話を再現するため、Python を使って人工無能の形態素解析やマルコフ連鎖について学び、自然な応答が可能なチャットボットを作成した。

## 2. 研究の具体的内容

### (1) Python とは

Python はプログラミング言語の一種で Web 開発やデータ分析など、汎用性が高く、人工知能などの先端技術の開発で人気の高い言語である。コードを書きやすく、読みなおしがしやすい点を踏まえて今回使用する言語とした。今回初めて Python を用いて作品を作ったが、C 言語が元になっている分、C 言語との違いに苦しむことが何度もあった。

### (2) なぜ人工無脳なのか

今回研究を行っていくうえで人工無脳と人工知能のどちらを作成するのかということになったが、人工知能には音声認識による会話を再現するうえで複雑なプログラムや応答に対する返答の正確性に欠けるという問題があり、人工無脳のほうがシンプルで開発しやすいという結論に至った。なお人工無能の名前はピティナにした。

### (3) ピティナの開発環境

ピティナを作成する上でたくさんのモジュールに分けて作成していくために、Anaconda Navigator の中の一つの開発ツールである Spyder を使って作成した。

### (4) オウム返しとランダム応答

ピティナとの会話を再現するために、最初にオウム返しとランダム応答という応答ができるところから作成していった。“オウム返し”とはこちらが話した内容を「!」や「?」などの語尾を追加して返答するプログラムで、処理の流れは図 1 に示す。”ランダム応答”とはこちらが話した内容に関係なく事前に用意した返答を返すプログラムで、処理の流れは図 2 に示す。

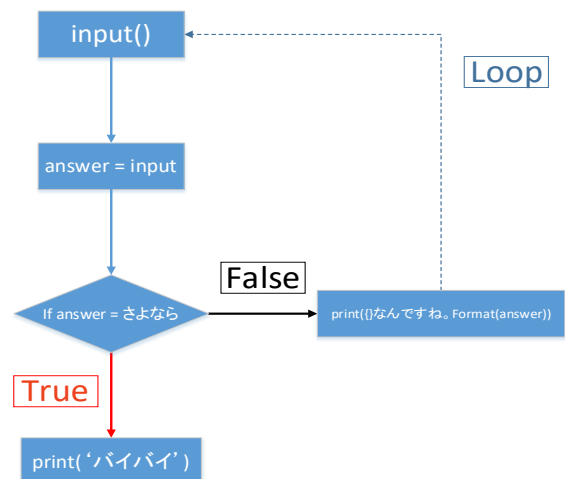


図 1 : オウム返し

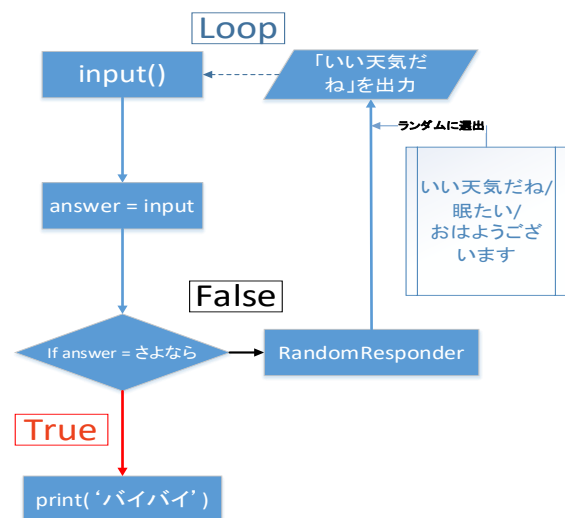


図 2 : ランダム応答

### (5) ピティナのUIの作成

“オウム返し”と“ランダム”が実装できた後、GUI作成を行った。作成にはAnaconda3にデフォルトで導入されている、「QtDesigner」というソフトウェアを使用した(図3)。ウィンドウの作成にはプログラムを手打ちするタイプがあるが今回使用したQtDesignerは視覚的にリソースと呼ばれるオブジェクトを配置して作成する直感的な作成が可能な方法になっている。必要な画像、ボタンなどをイベントハンドラと接続してクリックによる制御を可能にした。しかし、ここで制作しただけではただのUIファイルになってしまうので自分たちの扱うPythonで活用できるようにファイル形式を.uiから.pyに変える必要がある(図4)。

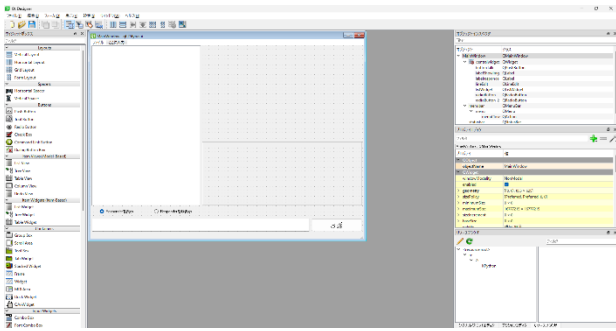


図3 : QtDesigner

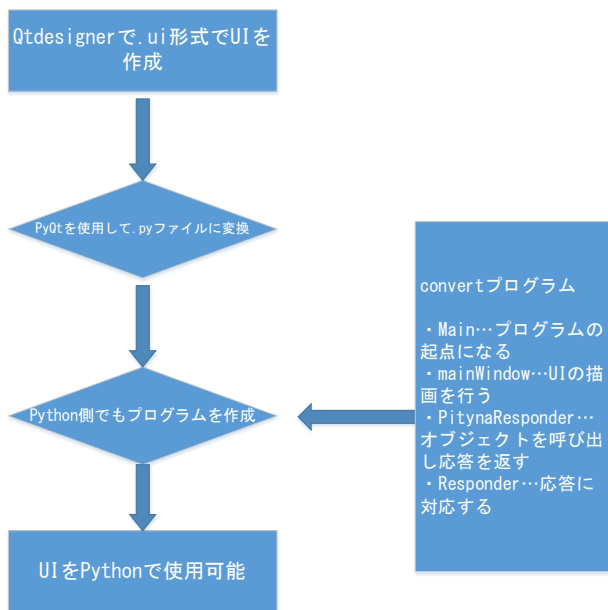


図4 : 変換方法

### (6) ランダム実行

最初に作成したオウム返しとランダム応答だが同時に実行するわけにはいかないのでランダムで選択して実行するようにする。1から100の値をランダムに算出することで処理が変化する。返答パターンが増える毎にこのプログラムは書き換える必要がある(図5)。

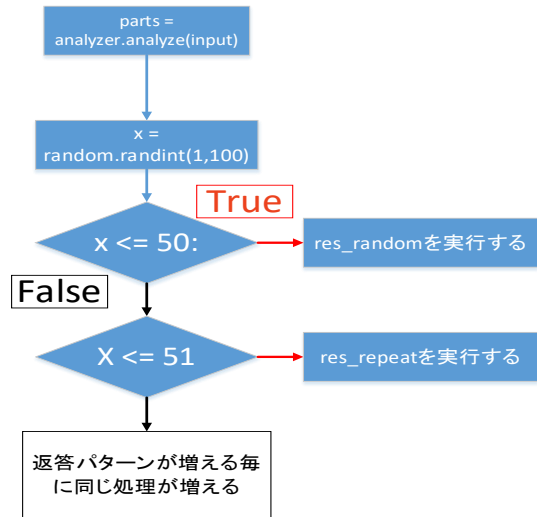


図5 : 返答パターンのランダム化

### (7) パターン応答の作成

オウム返しとランダム応答の次はパターン応答を導入していく。パターン応答はパターン辞書を作成してその辞書にマッチした会話をするすると決まった応答をしてくれるプログラムである。簡単に言うと「こんにちは」と入力すると事前に作成した辞書の「こんにちは」や「また会ったね」などの返答をする。図6のようにパターン辞書の作成はメタ文字という特殊な意味を持つ記号を使って作成するので書くときに少し癖がある。

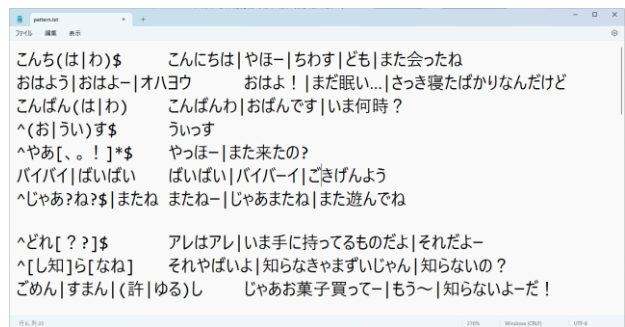


図6 : パターン辞書の例

### (8) 感情の創出

ピティナはプログラムなので私たちのように喜んだり悲しんだりすることはできないが、感情を持った「フリ」をさせることはできる。

プログラム内で感情を状態にして考えることで疑似的な感情を持たせることを実現した。「不機嫌⇄平常⇄上機嫌」の三つの状態に分けて「不機嫌」はマイナス「平常」は0「上機嫌」はプラスのように機嫌値を置く。そしてこの機嫌値を変化させることで疑似的な感情を作成することができる。そして機嫌値の値を特定の言葉が来た時に変化するようにする。これによって「フリ」ではあるが感情を作り出すことができる。

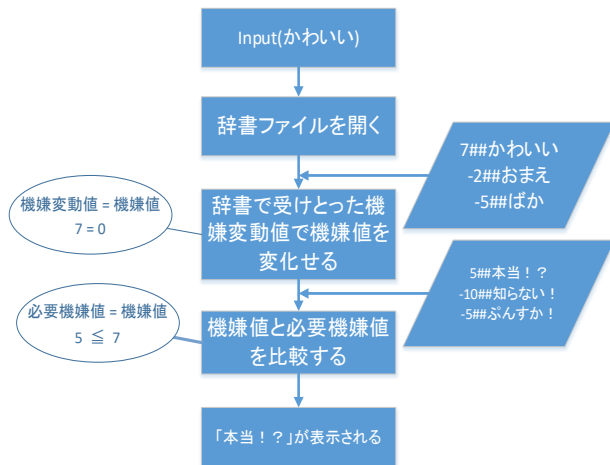


図7：機嫌値を使った返答方法

### (9) ピティナの機械学習

人工無能は人工知能と違って自分で考えて返答していない。これだと自然な会話は生み出せない。そこでピティナにも会話を「記憶」し「ピティナの言葉」で返せるよう機械学習能力を加えていった。まず、最初に作成したランダム応答用のファイル(random.txt)と新しく会話内容を保存するログファイル(log.txt)にそのままユーザの書いた会話が保存されるようにした。これによってランダム返答が行われた時ユーザの発言がそのまま返されるようになった。

### (10) 形態素解析

ログファイルに記憶できるようになったことでピティナはユーザの発言を覚えるようになった。だが今のままではただ言われたことを丸ごと返しているだけなので、文ではなく単語で会話を記憶することで応答フレーズがより自然になるようにした。文章を単語にするためには「形態素解析」という解析をする必要があり今回は Python のライブラリにある「Janome」を使用した。この形態素解析した単語をパターン辞書のパターン文字列として登録することで応答の時に解析した結果の応答が返ってくるようになった(図8)。

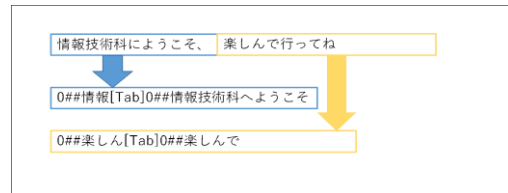


図8：形態素解析を使って文の作成

### (11) テンプレート応答

形態素解析は文章を解析して対応した応答を返すという学習をさせてきたが、次は逆に単語の部分を取り出した文章を作るという処理を作った。先ほどと同じように形態素解析を使うことで単語分けしその中の名詞の部分を取り抜き、これによってテンプレートの文章を作ることができるのである。テンプレートの文章を作ることができたらあとはピティナが単語を当てはめることで返答をすることができるようになった(図9, 図10)。

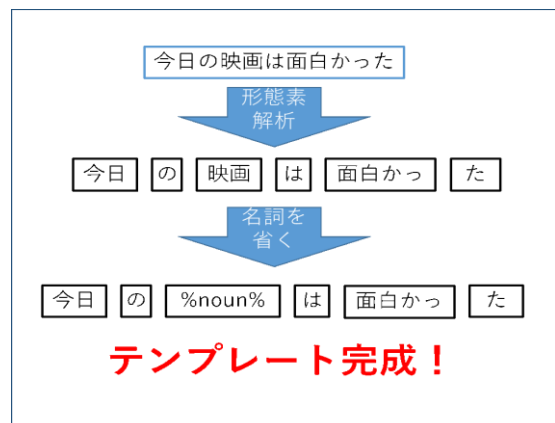


図9：テンプレートの作成

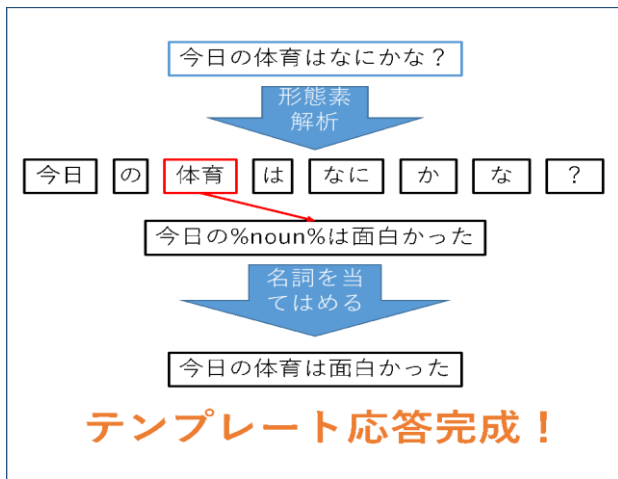


図 10：テンプレート応答の作成

### (12) マルコフ連鎖による応答の作成

マルコフ連鎖による作成はテンプレートなどの文章がある前提の文章の作成ではなく、ログファイル(log.txt)を読み取ることでいちから文章を作り出す方法である(図11)。ログファイルを形態素解析して単語にすることでマルコフ辞書を作成しマルコフ辞書から文章を作り出す。マルコフ連鎖の内容は簡単に説明すると確率によって単語を選別し、文章を作成していくというものである。

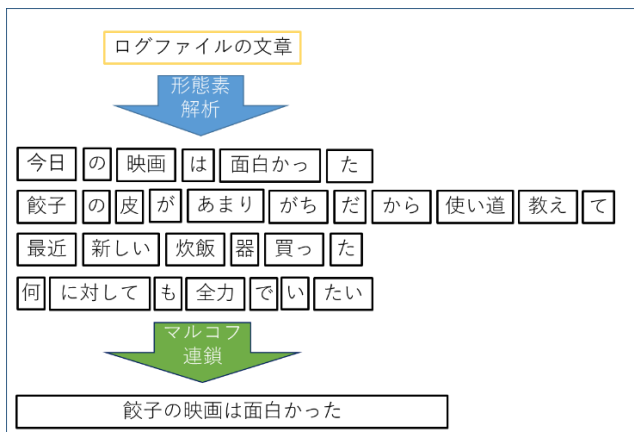


図 11：簡単なマルコフ連鎖の例

### 3. 研究のまとめ

今回の研究で人工無能を使った会話の実現を成功することができた。しかし、目標である音声認識を使った制御は実現することができなかった。反省点として、個人で作業をそれぞれ分けて行うことで効率を良くしようとしたが実際はそれぞれの進捗が噛み合わず思

ったような速度で作業を進めることができなかった。他にもプログラム面での問題もあるが研究に取り組む時間が少なかったことで満足した作品を作れなかったのはとても悔しく思う。AI技術が進歩する世の中少しでも人工無能の知識が生かせる機会があることを期待している。



図 12：ピティナの全体像

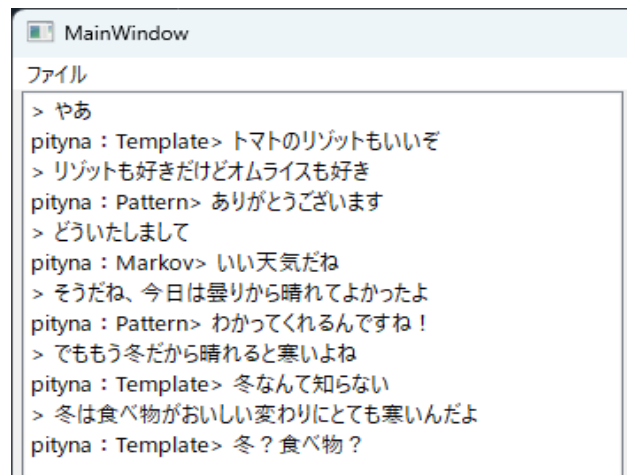


図 13：会話のログ

### 4. 参考文献

・[Pythonプログラミングパーフェクトマスター \[Python3/Anaconda/PyQt5 対応第3版\] - 秀和システム あなたの学びをサポート！ \(shuwasystem.co.jp\)](https://shuwasystem.co.jp/)

・平成30年度(2018)課題研究/Pythonを用いた人工無能の作成