

令和4年度 卒業研究

検索を利用した

AI チャットボットの作成

Creating AI chatbots using search

函館工業高等専門学校  
生産システム工学科 情報コース  
18番 白石 侃穩  
指導教員 東海林 智也

# 目次

第 1 章 序論	3
第 1 節 英文アブストラクト .....	3
第 2 節 研究背景 .....	3
第 3 節 研究目的 .....	3
第 4 節 開発環境 .....	4
第 2 章 プログラムの作成	5
第 1 節 概要 .....	5
第 2 節 ライブラリの機能 .....	5
第 3 節 完成したプログラム .....	9
第 3 章 まとめと今後の展望	11
参考文献・謝辞	12

# 第1章 序論

## 第1節 英文アブストラクト

In recent years, AI research has been active. Among them, most of the data used in interactive AI research is generic, and we started this research because we believe that using data obtained from users will enable more personalized conversations and bring us closer to achieving natural conversations. As a first step, we created a program that uses the chat function of LINE to provide an AI response and a program that displays frequently used websites based on the user's search history.

**Key words:** AI, Natural Conversation, personalize

## 第2節 研究背景

近年、AI技術によって社会にさまざまな変化が見られるようになり、人間を介さないサービスが数多く見られるようになった。そのようなサービスの一つに会話エージェントと呼ばれる「テキストや音声による対話を通じて人間のような会話をシミュレートするコンピュータプログラム」がある。現在使われている会話エージェントの例として人間の質問に回答する ChatGPT、人間がしたいことをアシストする Siri などがある。

この会話エージェントの技術はまだ発達途中であり、会話の論点がずれたり、予想していない返答が返ってきたりという問題点がある。そのような問題を解決するため、AIの学習データにどのようなものを使用するか、入力データ（質問）から出力データ（回答）をどのように推測するかなどの研究が進められている。

しかしこれらの研究は特定の個人を限定しない汎用的なデータを対象とするものが多いので、AIの学習の際に汎用的なデータではなく、あるユーザのパーソナルデータを使用すればより自然な会話を行える可能性があるのではないかと考えた。

## 第3節 研究目的

本研究ではユーザのパーソナルデータとして、そのユーザの検索履歴から推定した「ユーザの好み」を用いる。推定したユーザの好みを利用することでチャットボットとの会話がより自然なものになるかについて検証する。そのために LINE を利用して会話ができるチャットボットを作成する。

## 第 4 節 開発環境

本研究の開発環境は以下の通りである。

使用 OS : macOS

使用言語 : Python3.9.7

使用環境 : Visual Studio Code

主な使用ライブラリ : line-bot-sdk,[1] pychatGPT[2], mecab[3], janome, beautifulsoup, re

# 第2章 プログラムの作成

## 第1節 概要

本研究ではチャットボットを python により作成する。また LINE の Messaging API の機能を利用して LINE 上でチャットを行う。

## 第2節 ライブラリの機能

### (1) Line-bot-sdk

LINE の MessagingAPI のクライアントライブラリであり、このライブラリを用いると bot 開発を簡単に始められる。例えばイベント毎の処理を分けて記述したり、署名を検証する操作を簡単に書いたりすることができる。

使用例：

```
line_bot_api.reply_message(  
    event.reply_token,  
    TextSendMessage(text=texts))
```

text=の後に変数やメッセージを代入すると、それをリプライとして LINE サーバーに送りチャット画面に表示される。

### (2) re

python の標準ライブラリで、正規表現の処理を行うことができる。本研究では正規表現パターンによる文字列の抽出や置換をするために使用した。

使用例：

```
re.sub(r'&sa=.*', '', site_url)
```

URL に含まれる '&sa=' というパターンの後の文字列を消去している。第一引数に正規表現パターン、第二引数に置換後の文字列、第三引数に処理対象の文字列を指定する。

### (3) pychatGPT

OpenAI が提供している自然言語処理ライブラリである。人間のようなテキストを生成できるため、チャットボットなど幅広い用途で利用されている。本研究ではチャットの返答を生成するために使用した。

使用例：

```
response = openai.Completion.create(  
    model='text-davinci-003',  
    prompt=prompt,  
    temperature=0, # ランダム性の制御[0-1]  
    max_tokens=50, # 返ってくるレスポンストークンの最大数  
    top_p=1.0, # 多様性の制御[0-1]  
    frequency_penalty=0.0, # 周波数制御[0-2]: 高いと同じ話題を繰り返さなくなる  
    presence_penalty=0.0 # 新規トピック制御[0-2]: 高いと新規のトピックが出現しやすくなる  
)
```

上の例の様に、様々なパラメータを指定することができ、それによって、入力内容に対応する文章を自動生成することが可能である。文章の生成例を以下の図1に示す。

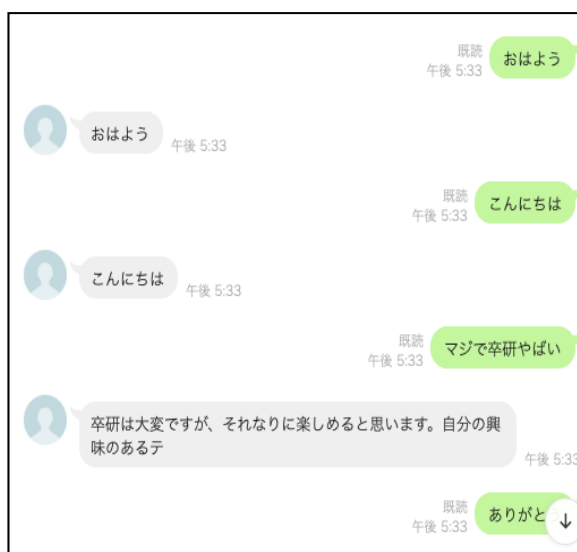


図1 pychatGpt の組み込み結果

#### (4) BeautifulSoup

Web スクレイピング用のライブラリで、HTML や XML ファイルの解析を行うことができる。本研究では google 検索の結果を解析するために利用した。

使用例：

```
# Google検索するキーワードを設定
search_word = 'python'

# 上位から何件までのサイトを抽出するか指定する
pages_num = 1 + 1

print(f'【検索ワード】{search_word}')

# Googleから検索結果ページを取得する
url = f'https://www.google.co.jp/search?hl=ja&num={pages_num}&q={search_word}'
request = requests.get(url)

# Googleのページ解析を行う
soup = BeautifulSoup(request.text, "html.parser")
search_site_list = soup.select('div.kCrYT > a')

# ページ解析と結果の出力
for rank, site in zip(range(1, pages_num), search_site_list):
    try:
        site_title = site.select('h3.zBAuLc')[0].text
    except IndexError:
        site_title = site.select('img')[0]['alt']
    site_url = site['href'].replace('/url?q=', '')
    # 結果を出力する
    print(site_title + " : " + site_url)
    print(url)
```

上の例では検索したいワードを指定して google 検索を行い、その検索結果の解析を行っている。図 2 にこの出力結果を示す。



図2 beautifulsoup の組み込み結果

## (5) mecab、janome

どちらも形態素解析を行うためのライブラリであり、文章を最小単位の単語に分解し解析を行うことができる。本研究では検索履歴の解析、ユーザから送られてきた文章の解析、google 検索の結果の解析等に利用している。

### 使用例 1:

ユーザから送られてきた文章に対して形態素解析を行って単語に分解した結果を図 3 に示す。

マジ	マジ	マジ	まじ	形状詞一般					1
で	デ	ダ	だ	助動詞	助動詞-ダ			連用形一般	
卒	ソツ	ソツ	卒	接尾辞-名詞的	一般				
研	ケン	ケン	研	接尾辞-名詞的	一般				
やばい	ヤバイ	ヤバイ	やばい	形容詞一般	形容詞	終止形一般			2
です	デス	デス	です	助動詞	助動詞-デス	終止形一般			
。				補助記号-句点					
助け	タスケ	タスケル		助ける	動詞一般	下一段-力行		連用形一般	3
て	テ	テ	て	助動詞	接続助詞				
ください	クダサイ	クダサイ		クダサル	下さる	動詞-非自立可能	五段-ラ行	命令形	3
EOS									

図3 mecab で形態素解析を行った結果



## 使用例 2 :

検索履歴から名詞、固有名詞を形態素解析により抽出し、それらの出現回数をカウントし頻出単語をとりだした結果を図 4 に示す。

```
[('Google', 254), ('YouTube', 81), ('ABEMA', 76), ('Python', 44), ('APIOpenAI', 29), ('com', 28), ('MV', 27), ('iOS', 24), ('OpenAI', 23), ('python', 22), ('DevelopersLINE', 19), ('Gmail', 19), ('LoginLINE', 19), ('re', 18), ('函館高専', 17)]  
('Google', 254)
```

図 4 mecab で検索履歴から頻出単語を抽出

## 第 3 節 完成したプログラム

完成したプログラムの実行画面を図 5、6 に示す。まずユーザの検索履歴を mecab と janome を用いて解析し、最も頻出する単語をユーザの好みとして推定する。さらに beautifulsoup を用いて推定したユーザの好みに関して google 検索を行う。その検索結果を返答することで、チャットボットがユーザとの会話で興味のある話題を提供する。なお pychatGPT を利用して自然な返答を行えるようになっている。



図5 完成したプログラムの実行画面

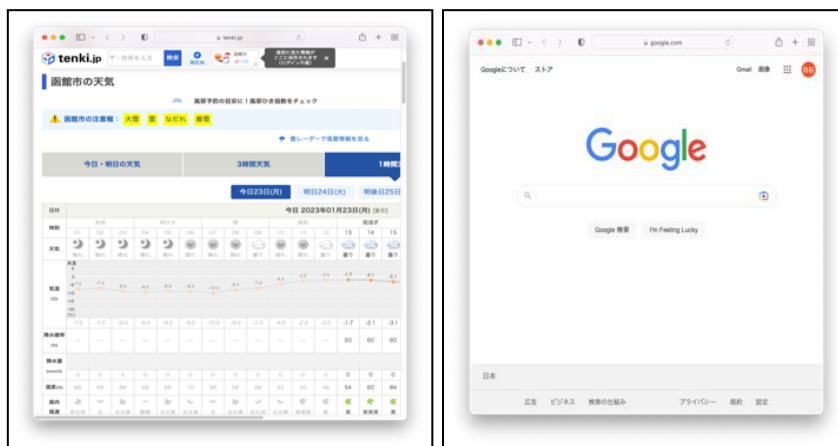


図6 完成したプログラムの実行画面

## 第3章 まとめと今後の展望

今回は検索履歴の中で最も頻出する単語をそのユーザの好みとしていたが、それではまだユーザの好みを的確に推定できるとは言えない。そこで今後は、ひとつだけでなく頻出単語の上位数個を利用する、検索履歴以外のデータを利用するなど、ユーザの好みを正確に推定できるような工夫をおこなう。また、不自然な返答を減らし、より人間らしい会話ができるようにする。

なお今回は研究背景で述べた「AIの学習の際に汎用的なデータではなく、あるユーザのパーソナルデータを使用すればより自然な会話を行える可能性があるのではないか」について検討することが出来なかった。そこで今後は「ユーザの好み」以外のパーソナルデータについても考慮し、この問題に関する検討をおこなう。

# 参考文献・謝辞

## 謝辞

本研究を進めるにあたりご指導くださいました、東海林智也准教授に感謝いたします。

## 参考文献

[1] LINE Developers

<https://developers.line.biz/ja/>

[2] Openai

<https://openai.com/blog/chatgpt/>

[3] Mecab 公式

<https://taku910.github.io/mecab/>