

# 情報理工学特別研究報告書

## 題目

ChatGPT を用いた英会話学習に関する研究

学生証番号 053697

氏名 谷掛 優太

提出日 令和6年1月26日

指導教員 蚊野 浩

京都産業大学  
情報理工学部

## 要約

世界のグローバル化に伴い、英会話を習得することへの需要が高まっている。英語を第二言語として習得できると、就職や進学に有利である、留学の際に支援を受けやすい、海外旅行をより楽しむことができる、などのメリットがある。しかし、英会話を習得するためには様々な困難が存在する。本研究では、ChatGPT を使用して、英会話学習を楽に継続できるシステムを開発した。

本研究では、音声認識を行う Python のモジュールである SpeechRecognition を用いて発話者の発言をテキストデータに変換し、変換したテキストデータを ChatGPT に送信する。その後、ChatGPT で生成された返答をテキストデータとして受け取り、そのテキストデータを、音声合成を行う Web API である gTTS を用いて音声データに変換する。変換した音声データをスピーカーから再生することで発話者への返答とする。この一連の動作を繰り返すことで ChatGPT との英会話を行う。この基本機能に加えて、ChatGPT に対するプロンプトによる制御や、発話者の感情を入力するツールを追加することで、会話に抑揚を持たせるようにした。

システムを構築し、テストした結果、プロンプトによる制御を行わずに英会話を行うと、ChatGPT が生成する返答が長くなり、一方的な会話になる傾向があった。プロンプトによる制御を加えて返答の長さを抑えると、会話の自然さが改善された。さらに感情を入力することで ChatGPT に発話者の現在の感情が送られるようになり、会話に抑揚を持たせやすくなった。

これらの結果により、英会話学習を楽しくするという目標に近づいたが、いくつか課題も残った。ChatGPT からの返答を待つ間や、生成された返答を音声データに変換する間のラグがやや目立ってしまった。この問題は生成された返答を全て同時に音声変換するのではなく、生成された部分から音声データに変換することで解決できそうであった。また、現状のシステムでは UI のボタンを押して感情を入力しているが、実際は表情を読み取り、自動的に感情を入力するシステムを構築したかったのでそのシステムを完成させることができなかった点も課題点である。

## 目次

1 章 序論	．．． 1
2 章 ChatGPT API と text-to-speech, speech-to-text ライブラリ	．．． 2
2.1 ChatGPT と ChatGPT API	．．． 3
2.2 ChatGPT のプロンプト	．．． 7
2.3 speech_recognition	．．． 8
2.4 gTTS	
3 章 ChatGPT と英会話を行うためのプロトタイプ制作	．．． 9
3.1 ChatGPT API の導入	．．． 9
3.2 speech_recognition と gTTS ライブラリの実装	．．． 12
3.3 プロトタイプの概要	．．． 13
3.4 プロトタイプの構築	．．． 13
3.5 感情を入力するための機能の追加	．．． 14
4 章 プロンプトと追加機能の比較実験	．．． 16
4.1 プロンプトの制御なしで英会話を行う	．．． 16
4.2 プロンプトによる制御を加えて英会話を行う	．．． 17
4.3 感情を入力する機能を追加して英会話を行う	．．． 18
5 章 結論	．．． 20
参考文献	．．． 21
謝辞	．．． 21
付録	．．． 22

## 1 章 序論

世界のグローバル化に伴い英会話への需要が高まっている。英語を母国語として使っている人は4億人程度であるが、第二言語として使用している人は非常に多く、世界中で約17億人いると言われている。英会話ができることで、就職や進学で有利である、留学の際に支援を受けやすくなる、海外旅行をより楽しむことができる、などのメリットがある。

英会話学習を行う方法はたくさんある。しかし、学習を継続し、英会話を習得することは容易ではない。その理由は、英会話学習に面白みを感じられない、英会話はすぐに身につかない、自分のレベルにあった学習ができない、などである。これらの理由により、英会話を習得しようとしても、思うように学習が進まない、継続した学習ができずに習得を諦めてしまうことが多いように思われる。

昨今、世界中でユーザー数が増加している ChatGPT という人工知能チャットツールが存在する。ChatGPTは大量のテキストデータを学習しており、質問した内容に対し返答する、論文などの長い文章を短時間で要約する、プログラムのソースコードを記述するなど様々な使用方法がある。チャットボットとしての機能も備えているため、ChatGPTを相手に、日常会話を行うことも可能である。ChatGPTはOpenAIというアメリカのカリフォルニア州にある会社が開発したものであるため、英語の学習データが非常に多い。そのため、チャットボットとして使用する際には、英語を使用するとより使いこなすことができる。

本研究では ChatGPT の会話機能に注目した。ChatGPT を用いて英会話学習を行うことで、英会話学習を困難にしている要因の一部を解決することが目的である。そのために ChatGPT と英会話するためのシステムを Python で開発した。以下、2章ではシステムを開発するために使用したツールに関して、3章ではシステムの構築について、4章では実験結果について記述する。5章では結論について記述する。

## 2章 ChatGPT API と text-to-speech, speech-to-text ライブラリ

2章では, ChatGPT を用いて英会話を行う具体的な方法と, そのために使用したツールである ChatGPT API, gtts, speech\_recognition を説明する.

### 2.1 ChatGPT と ChatGPT API

ChatGPT は, OpenAI によって開発された人工知能ベースの対話システムである. ChatGPT は大量のテキストデータから学習を行い, 人間のように会話を行う能力を持っている. また, ChatGPT は質問に答えたり, 話題についての情報を提供したり, 文章の作成を行うこともできる. 加えて, プログラムのソースコードの作成や修正, 簡単な数学の問題を解くこともできる.

ChatGPT は自然言語処理システムを利用している. これにより, 人間の言葉を理解し, 自然な返答を生成することができる. しかし, このシステムは完璧ではなく, 誤った情報や不正確な情報を提供することもある. その理由の一つは, ChatGPT の知識は特定の時点のものであるため, 最近の出来事やトレンドに関する情報が不足しているためである.

現在 ChatGPT は, 教育, エンターテイメント, ビジネス, 技術サポートなど多岐にわたる用途で利用されている. ChatGPT のユーザーは, テキストベースのインターフェースを通じて ChatGPT にアクセスし, さまざまな質問や要求を行うことができる. このツールは, 知識の提供, アイデアの生成, 教育的な目的, コミュニケーションのサポートなど, 様々な方法で役立てられている. 以下の図は実際に ChatGPT を使用した例である.

**You**  
京都について教えてください

**ChatGPT**  
京都は日本の関西地方に位置する都市で、日本の古都としても非常に有名です。長い歴史と文化、そしてその美しい自然景観で知られています。以下に、京都の特徴をいくつか紹介します：

### 歴史

- 京都は794年に平安京として建都され、約1000年間にわたって日本の首都でした。
- 多くの歴史的な建造物や文化財が残っており、これらは日本の歴史や文化を理解する上で非常に重要です。

### 文化

- 京都は日本の伝統文化が色濃く残る都市であり、伝統的な工芸品、茶道、華道、日本舞踊などが今もなお盛んに行われています。
- 祇園祭や葵祭、時代祭などの伝統的な祭りも年間を通じて開催され、多くの観光客を惹きつけています。

### 観光

- 京都は観光都市としても有名で、金閣寺、清水寺、嵐山、伏見稲荷大社など、多くの有名な観光スポットがあります。
- 世界遺産にも複数登録されており、古い町並みや美しい庭園などが人気です。

### 自然

- 四季折々の美しい自然に囲まれており、特に桜や紅葉の季節は多くの人々を魅了します。

## 図 2.1 ChatGPT を使用した様子

ChatGPT API は、ChatGPT の機能を呼び出して利用することができるプログラミングインターフェイスであり、OpenAI 社から提供されている。この API を使用することで、ChatGPT が学習したデータベースに、他のプログラムからアクセスし、ユーザー体験を向上させたり、新しいアプリケーションの機能を開発したりすることができる。例えば、カスタマーサポートチャット、教育用アプリ、などを作ることができる。

## 2.2 ChatGPT のプロンプト

ChatGPT のプロンプトとは、ユーザーが ChatGPT に対して入力するテキストのことを指す。プロンプトは、一般には、質問、命令、話題の提案、または単なるコメントなどであるが、さまざまな形式で入力することが可能である。ChatGPT に何を生成してもらいたいかを事前に明確に伝えることで、希望に沿った内容の回答を得ることが可能になる。

プロンプトの内容や形式は多岐にわたるが、ここでは主たるものとして、4 つのタイプを紹介する。

- ① 質問型:ユーザーが何かを知りたい場合に,具体的な質問をプロンプトとして使用する.例として ChatGPT に対して「京都の観光名所はどこですか?」と質問した回答を図 2.2 に示す.

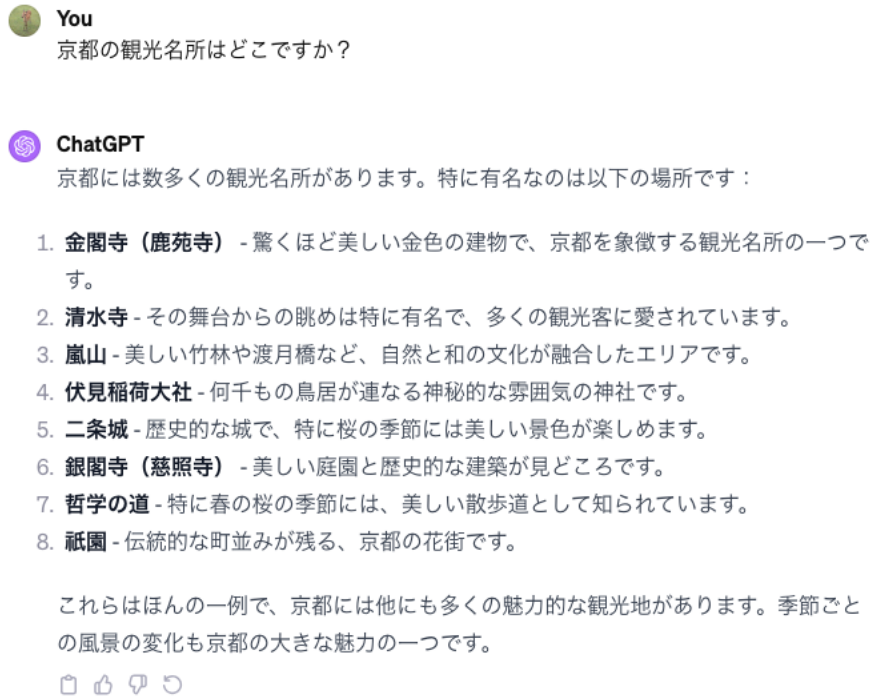


図 2.2 質問型プロンプトの使用例

- ② 命令型：ユーザーが ChatGPT に特定のアクションを実行するように指示するプロンプトである。図. 2.3 では、ChatGPT に昔話「桃太郎」の要約を命令している。

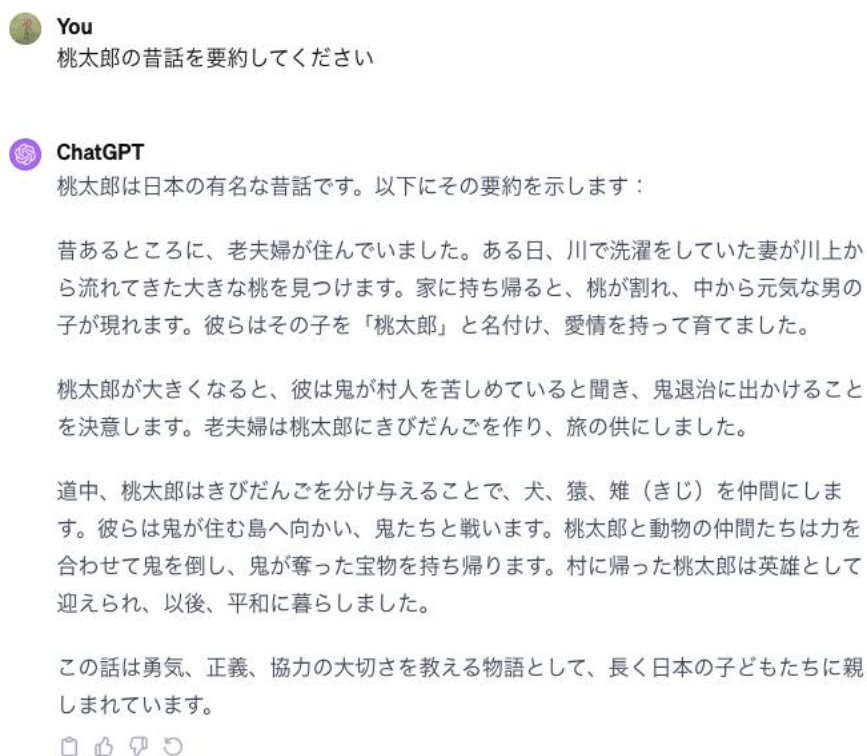


図 2.3 命令形式プロンプトの使用例



- ③ 対話型：ユーザーが ChatGPT と対話形式のコミュニケーションを行うプロンプトである。図 2.4 は、ChatGPT に「今日は読書をして過ごそうと思う」、「夏目漱石の本を読もうと思う」と入力し対話型のコミュニケーションを行った様子である。

**You**  
今日は読書をして過ごそうと思う

**ChatGPT**  
読書で過ごす日というのは素晴らしいですね。本を読むことは、知識を得たり、想像力を豊かにしたり、リラックスするのに最適な方法です。今日はどのような本を読む予定ですか？小説、ノンフィクション、詩集、自己啓発書など、ジャンルによっても読書体験は大きく異なります。

もしまだ読む本を決めていない場合、あなたの興味や現在の気分に合わせた本の推薦もできますよ。例えば、冒険やファンタジーが好きなら、そのジャンルの名作を、落ち着いたものが読みたい場合は古典文学や現代の文学作品を提案することができます。

また、読書をさらに楽しむための小さなヒントもあります。例えば、静かな場所を選んだり、お気に入りの飲み物を用意したり、快適な読書スペースを作ることで、より読書に没頭できるかもしれません。

読書中に興味深いテーマや疑問点が出てきたら、それについて話し合うのも楽しいですよ。もし読書に関する何か質問があれば、いつでも尋ねてください。リラックスした読書の時間をお楽しみください！

**You**  
夏目漱石の本を読もうと思う

**ChatGPT**  
夏目漱石の本を読むというのは素晴らしい選択ですね。夏目漱石は日本文学の巨匠であり、彼の作品は今なお多くの読者に愛されています。漱石の作品は、深い人間洞察、独特なユーモア、そして時に哲学的な思索を含んでおり、日本文学を代表する作品として高く評価されています。


漱石の代表作には以下のようなものがあります：


- 『吾輩は猫である』：一匹の猫を主人公にしたユーモラスで風刺的な作品。日本社会や人間の愚かさを猫の目を通して描いています。
- 『坊っちゃん』：教師として四国の学校に赴任した主人公の奮闘を描いた作品。正義感が強いが頑固な主人公のキャラクターが魅力的です。
- 『こころ』：師弟関係や友情、恋愛を通じて、人間の心の複雑さを深く掘り下げた作品。漱石の晩年の代表作です。

これらの作品は、日本の近代化とその影響を受けた人々の心理を巧みに描き出しており、現代の読者にも多くの洞察を与えます。また、漱石の文体は、彼の豊かな教養と洗練されたユーモアセンスを反映しており、読む人を魅了します。

## 図 2.4 対話型プロンプトの使用例

- ④ クリエイティブ型：ユーザーが ChatGPT に物語やストーリーを生成したり詩や川柳などの創作物を生成したりするために指示するプロンプトである。図 2.5 では、森の中に住むクマをテーマに物語を生成させている。

 **You**  
森の中に住むクマをテーマに物語を始めてください

 **ChatGPT**  
かつて、深い緑に覆われた広大な森の中に、一匹の熊が住んでいました。この熊はバルーと呼ばれ、森の生き物たちから深い敬意を受けていた。彼はその大きな体と温かい心で、森の住人たちの悩みを聞き、力になっていました。

ある日のこと、バルーがいつものように木の実を集めていると、小さな鳥が慌てて飛んできて、「森の北の方に何か変わったことが起こっている！」と伝えました。バルーはすぐに、何が起こっているのか確かめに行くことにしました。

森の北へと進むにつれ、バルーは徐々に変化に気づき始めました。いつもなら聞こえるはずの鳥のさえずりや小動物たちの声が聞こえない。そして、空気には何とも言えない異変の匂いが漂っていました。

やがてバルーが森の北端に到達すると、目の前に広がっていたのは、見たこともないような不思議な光景でした。そこには、大きな石と謎めいた光があり、そして...

ここから物語は、バルーが森の新たな秘密を解き明かす冒険へと進んでいきます。彼は森の平和を守るため、そして森の住人たちの未来のために、未知の挑戦に立ち向かうのでした。



## 図 2.5 クリエイティブ型プロンプトの使用例

自分の目的に合わせてこれらのプロンプトを選択することで、ChatGPT はさまざまな仕事の手助けを行う。今回の研究では対話型を使用する。

### 2.3 speech\_recognition

Python の `speech_recognition` ライブラリは、音声データをテキストに変換するためのツールである。このライブラリは、ファイルやマイクロフォンなど、様々なオーディオソースからの音声データをテキストデータに変換することができる。また音声データ処理の際にノイズを自動で調整し、音声認識の精度を向上させることができる。

`speech_recognition` ライブラリは、自動字幕生成、音声コントロールシステム、対話型ボット、言語学習アプリケーションなど、多岐にわたる分野で活用されている。特に、音声ユーザーインターフェースが重要な役割を果たすアプリケーションでその価値が認められている。

## 2.4 gTTS

Python の gTTS ライブラリは, Python でテキストデータを音声に変換するためのツールである. Google のテキスト読み上げ技術を使用して, 様々な言語とアクセントでテキストから自然な音声を生成することができる.

主な特徴としては以下の点が挙げられる.

- ① 多言語をサポート: gTTS は 40 以上の言語をサポートしている.
- ② 読み上げ速度の調整: 本研究では使用していないが, テキストの読み上げ速度を調整することができ, ユーザのニーズに合わせた音声を生成することができる.
- ③ MP3 ファイル出力: 生成した音声を MP3 形式で保存することができる.

これらの特徴から gTTS ライブラリは, アプリケーションやウェブサイトでのアクセシビリティ向上, 言語学習ツール, オーディオブックの作成, AI ボットや対話型システムでの応答生成などに使用することが可能である.

gTTS ライブラリは Google のサービスを利用しているため, 使用する際にはインターネット接続が必要な点に注意する必要がある. また, 生成する音声が高い場合には音声の出力に時間が掛かる点にも注意する必要がある.

### 3章 ChatGPT を用いた英会話ツールの開発

ChatGPT を使用して英会話を行うために、音声を用いて会話を行うプロトタイプを開発した。以下の環境を利用した。

開発環境： Macbook Air Retina, Visual studio code  
プログラミング言語： Python3.8.8  
使用した API： ChatGPT API  
使用した python ライブラリ： speech\_recognition, gTTS

#### 3.1 ChatGPT API の導入

Python から ChatGPT の機能を利用するために ChatGPT API を導入する。本節では ChatGPT API の具体的な導入方法を説明する。まず事前に OpenAI ライブラリをインストールする。OpenAI ライブラリは以下のコマンドをターミナルに入力することでインストールされる。

```
$pip install speech_recognition
```

次に、OpenAI のサイトでアカウントを作成する。アカウント作成後、図 3.1 の右上にある Personal アイコンをクリックする。クリックするとタブが開かれるのでタブ内にある「View API keys」を選択する。

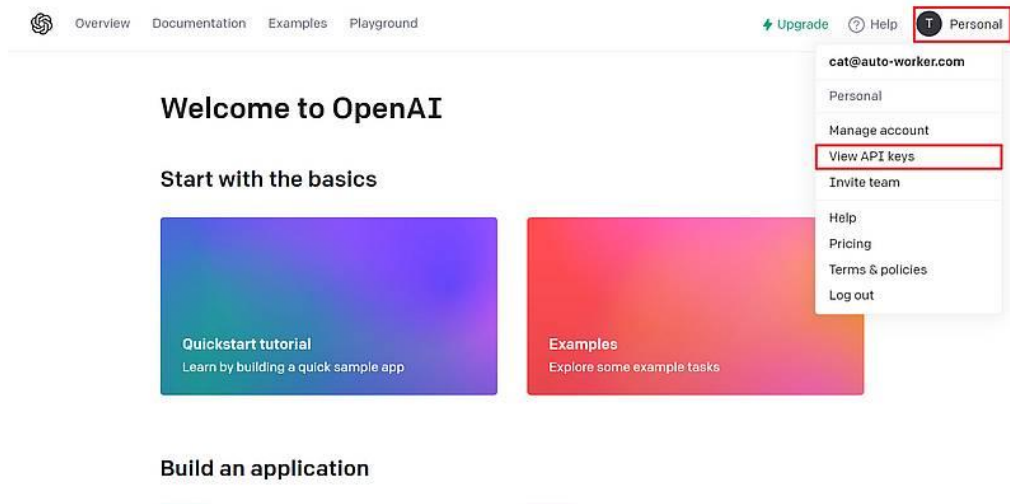


図 3.1 OpenAI の公式ホームページで View API keys を選択する様子

図 3.2 の「API keys」と表示されたページが表示されるので、中央にある「+ Create new secret key」 ボタンをクリックする。

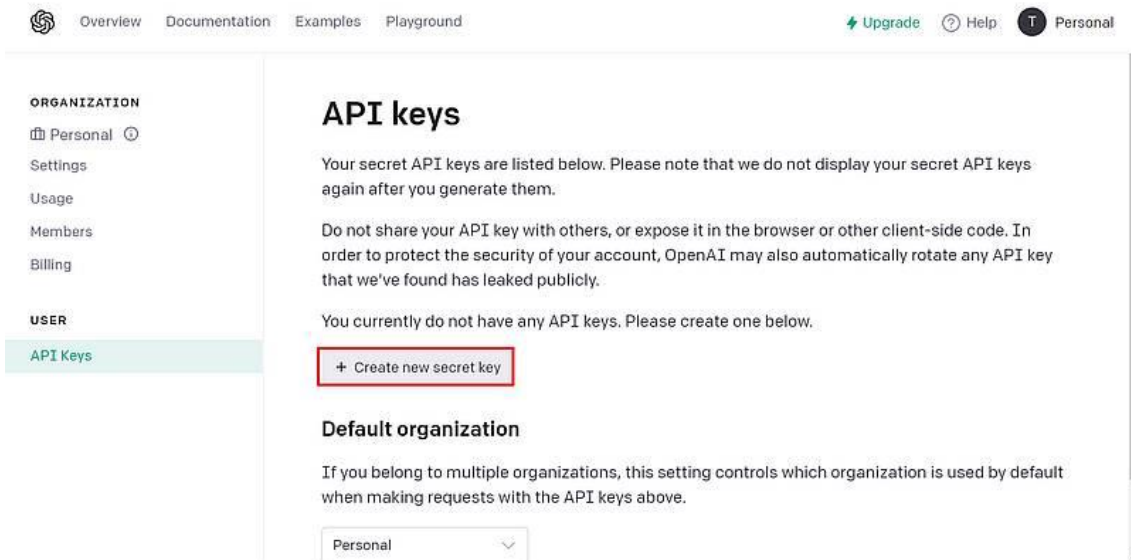


図 3.2 OpenAI のサイトで API を取得する様子

図 3.3 のウィンドウが開くので、赤枠アイコンをクリックしコピーすることで、そこに表示されている API キーを取得する。

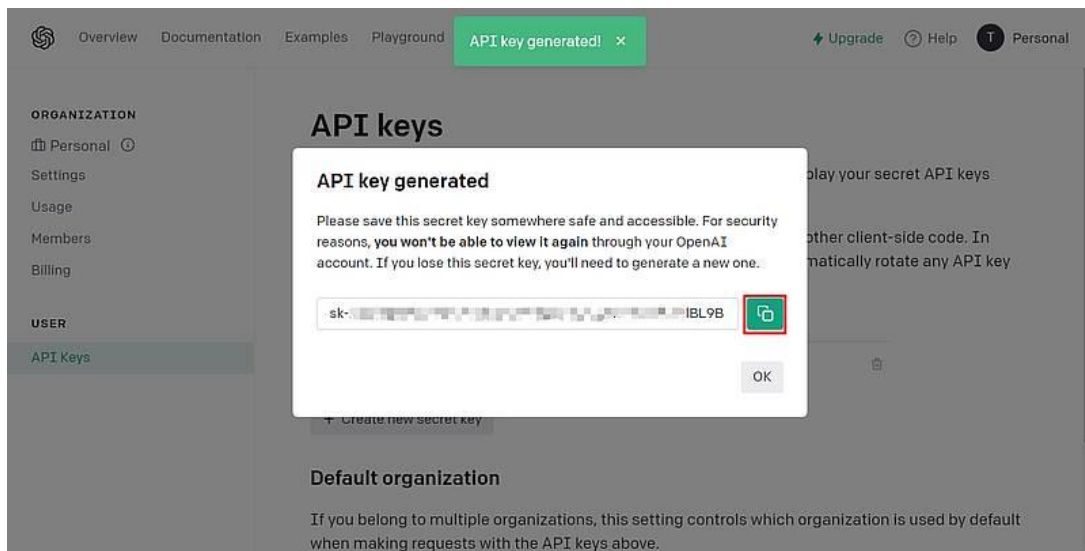


図 3.3 生成された API キー

以上の作業が API キーを取得する手順である。セキュリティ上の観点から API キーは他人に見せてはいけなないので、図 3.3 では API キーを隠している。取得した API キーを図 3.4 のようなサンプルコードで実行することで使用することができる。このサンプルコードでは ChatGPT に料理を生成する役割を 9 行目で与え、豚肉とネギを使ったメニュー

一を提案するように 10 行目で指定している.

```
1 import openai
2
3 openai.organization = "Organization IDをここに入れる"
4 openai.api_key = "API keyをここに入れる"
5
6 response = openai.ChatCompletion.create(
7     model="gpt-3.5-turbo",
8     messages=[
9         {"role": "system", "content": "あなたは30分以内に作れる夕飯メニューを生成するAIです。"},
10        {"role": "user", "content": "豚肉とネギを使ったメニューを箇条書き形式でほしい。"},
11    ],
12)
13
14 print(response['choices'][0]['message']['content'])
```

図 3.4 ChatGPT API を使用したサンプルコード

### 3.2 speech\_recognition と gTTS ライブラリの実装

本節ではプロトタイプで音声認識とテキストを音声変換するために使用した2つのライブラリについて実装方法を説明する。

Mac に 搭載されているターミナルを開く. 以下のようにコマンドを入力する.

```
$pip install speech_recognition
```

```
$pip install gTTS
```

これらを入力し実行することでライブラリがインストールされる. 次にこれらのライブラリが正しくインストールされていることを確認するために, 以下の二つのソースコードを実行する。

```
1 import speech_recognition as sr
2
3 AUDIO_FILE = "sample.mp3"
4
5 # use the audio file as the audio source
6 r = sr.Recognizer()
7 with sr.AudioFile(AUDIO_FILE) as source:
8     audio = r.record(source) # read the entire audio file
9
10 result=r.recognize_google(audio, language='ja-JP')
11
12 print(result)
```

図 3-5 speech\_recognition ライブラリの動作を確認するためのサンプルコード

```
1 from gtts import gTTS
2
3 s = 'これはサンプル音声です。'
4
5 tts = gTTS(s, lang='ja')
6 tts.save('gTTS_test.mp3')
```

図 3-6 gTTS ライブラリの動作を確認するためのサンプルコード

これらのサンプルコードを実行すると speech\_recognition の動作確認では, 用意した任意の音声ファイルの中身がテキストとして表示される. gTTS の動作確認では, 音声データが生成され, 生成された音声データを再生すると日本語音声で「これはサンプル音声です」と再生される。

### 3.3 プロトタイプの概要

本研究で作成したプロトタイプは以下の手順で動作する。

- ① 発話者がマイクから入力した音声を, `speech_recognition` ライブラリを用いて音声認識させる。
- ② 音声認識したデータをテキストデータに変換する。
- ③ テキストデータを ChatGPT に送信する。
- ④ ChatGPT が生成した返答をテキストデータとして受け取る。
- ⑤ 受け取ったテキストデータを, `gTTS` ライブラリを用いて音声データに変換する。
- ⑥ 変換した音声データをスピーカーから再生する。
- ⑦ ①～⑥までの作業を繰り返す。

### 3.4 プロトタイプの構築

3.1 節で導入した ChatGPT API, 3.2 節と 3.3 節で紹介した `speech_recognition`, `gTTS` を組み合わせて一つのシステムとして構築した python プログラムである, `sample.py` を開発した。このファイルをターミナルから実行することで英会話を行うことができる。そのプログラムの一部を図 3.7 に示す。

プログラムでは, 3.3 節で紹介した一連の動作を `while` 文として一つにまとめている。加えて `speech_recognition` と `gTTS` の処理は関数として, 149 行目の `recognize_speech()` 関数を呼び起こすことで一連の動作を実行している。

151 行目からの `if` 文の中では主に ChatGPT の設定を行なっている。154 行目の `response` の中身は `role` が発話者であるか, ChatGPT であるかを表しており, `contents` は会話の内容を表している。155 行目の `conversationHistory` ではこれまでの会話の履歴の格納を行なっている。

163 行目では API からの返答データにはメタデータなどが含まれており, 全ての返答を表示すると文字化けを起こす可能性があるため, 回答部分のみを抽出するように指示している。



```

147 while True:
148     # 音声認識関数の呼び出し
149     text = recognize_speech()
150
151     if text:
152         print(">> Waiting for response from ChatGPT...")
153         # ユーザーからの発話内容を会話履歴に追加
154         user_action = {"role": "user", "content": text}
155         conversationHistory.append(user_action)
156
157         # ChatGPTからの応答を取得
158         response = openai.ChatCompletion.create(
159             model="gpt-4",
160             messages=conversationHistory,
161             #max_token=75
162         )
163         response = response["choices"][0]["message"]["content"]
164
165         # ChatGPTからの応答内容を会話履歴に追加
166         chatGPT_response = {"role": "assistant", "content": response}
167         conversationHistory.append(chatGPT_response)
168
169         print("[ChatGPT]") # 応答内容をコンソール出力
170         print(response.strip()) # 応答内容をコンソール出力
171         # # (step3) 音声合成関数の呼び出し("ChatGPTのレスポンス"を"mp3ファイル"に変換して再生)
172         print(">> Generating audio file...")
173         text_to_speech(response)

```

図 3.7 開発したソースコードの一部

### 3.5 感情を入力するための機能の追加

この研究の「英会話を楽しくする」という目標を達成するために機能の追加を行なった。追加機能の内容は発話者の表情から感情を読み取り、読み取った表情に応じた感情を入力することで会話に抑揚をつける、というものである。感情を読み取らせることで、今までの、一定のトーンで単調な会話に、声の高低や感情の変動が生まれ会話が楽しくなるという考えのもと、この機能を追加した。

しかし、人の表情からカメラを通して感情を読み取るという機能を実装することが難しかったので、その前段階として図 3.8 のようなユーザーインターフェイスを追加し、ユーザがボタンを押すことで自分の感情を入力できるようにした。

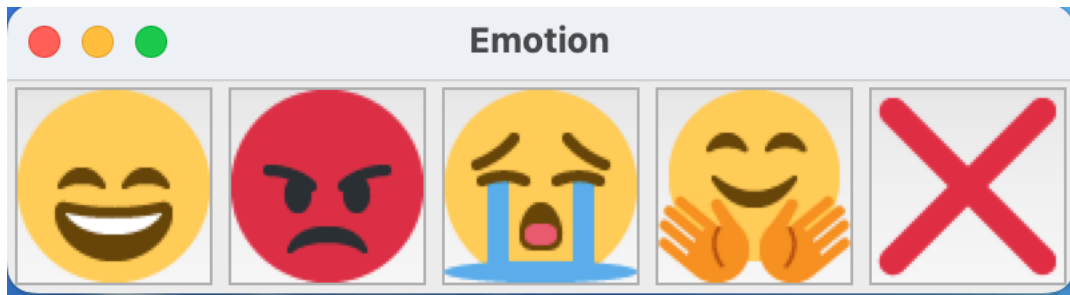


図 3.8 追加したユーザーインターフェイスの様子

このボタンは人間の喜怒哀楽に対応している。使い方は、会話を発話する前に、その時の感情に応じたアイコンを押す。そうすることで、ChatGPT との会話の中に、感情に対応する言葉が入力される。図 3.9 はこの機能を使って会話を行なった様子である。ここでは左から 3 番目の「悲しみ」を表すアイコンを押して会話を行なっている。

```
emotion:Sorrowful.  
>> Please speak now...  
[You]  
I practice tennis for competitions  
>> Waiting for response from ChatGPT...  
[ChatGPT]  
I'm sorry to hear that you're feeling this way. Tennis can be tough, especially when  
you're preparing for competitions. It's normal to feel stressed but remember, it's  
also important to enjoy the game.  
>> Generating audio file...  
>> Ready to Sppeach!
```

図 3.9 感情を入力して会話を行なっている様子

「私は大会のためにテニスの練習をしている。」という入力に「悲しい」という感情を加えた。この入力に対して、「このようなお気持ちをお察しします。テニスは厳しいもので、特に大会の準備をしているときは大変です。ストレスを感じるのは普通のことですが、試合を楽しむことも大切だということを忘れないでください。」と返している。これにより入力した感情に対して対応した返答を返していることがわかる。

## 4 章 プロンプトと追加機能の比較実験

この章では、第3章で開発したプロトタイプを用いて行なった実験と実験結果を示す。実験はプロンプトでの制御を行わずに会話を行なった場合、プロンプトでの制御をしつつ会話を行なった場合、プロンプトでの制御と感情を表現する機能で感情を入力して会話を行なった場合の3つの条件で行なった。

### 4.1 プロンプトの制御なしの英会話

まず、ChatGPT に対しプロンプトによる制御を設けずに英会話を行なった。会話に使用した英語は以下の3つである。

- We're almost at the top of the mountain.
- Great! I want to get a good picture.
- Let's have some lunch first.

図 4.1 は実際に行なった会話の様子である。

```
we are almost at the top of the mountain
>> Waiting for response from ChatGPT...
[ChatGPT]
That's fantastic! The last stretch can be the most challenging, but also the most rewarding. Remember to conserve your energy, stay hydrated, and keep a steady pace. The view from the top will be worth it!
>> Generating audio file...
>> Ready to Speeach!
>> Please speak now...
[You]
great I want to get a good picture
>> Waiting for response from ChatGPT...
[ChatGPT]
Absolutely! A picture from the top of the mountain would be an amazing memory to have. Just make sure to be safe while capturing your shot - don't get too close to the edge and always watch your step. Safety comes first, but I'm sure you'll get a breathtaking photo!
>> Generating audio file...
>> Ready to Speeach!
>> Please speak now...
[You]
let's have some lunch fast
>> Waiting for response from ChatGPT...
[ChatGPT]
Sure, let's get that lunch out. Eating something will give us the energy we need for the final push. I hope we've got something delicious packed! Remember to also drink plenty of fluids to stay hydrated.
```

図 4.1 プロンプトによる制御なしで英会話を行なった様子

会話として十分成立しているが、ChatGPT からの返答が長くなってしまっている。

## 4.2 プロンプトによる制御を加えた英会話

次にプロンプトによる制御を加えて英会話を行なった。加えたプロンプトは以下の通りである。

- Let's practice English conversation!
- Your name is Jenny. You should speak only as Jenny.
- Each conversation should be no more than 80 words.
- Please use CEFAR A2 level English.

“Let's practice English conversation!” は、これから英会話を行うという命令形式のプロンプトである。

“Your name is Jenny. You should speak only as Jenny.” は、相手に“Jenny”という役割を与え、それ以外の発言をしないように制御している。会話以外の発言をしないようにしている理由は、制御なしの状態では会話を行うと、発話者の発言に表現の間違ひがある場合に修正を行ったり、他の表現の提案をしてしまったりするためである。

“Each conversation should be no more than 80 words.” は、文字数の制限を行わないと 4.1 節のように返答が長くなってしまうためである。また、感情を入力した際に、短すぎると会話に反映できないことが頻出したため、80 ワードに調整している。

“Please use CEFAR A2 level English.” は、英語表現の難易度を下げるために使用している。

次のページ冒頭の、図 4.2, 図 4.3 は実際に会話を行なった様子である。

```
[You]
we are almost at the top of the mountain
>> Waiting for response from ChatGPT...
[ChatGPT]
That's so exciting! It's been tough, but reaching the top is going to feel amazing. Let's keep
pushing just a little bit more!
>> Generating audio file....
>> Ready to Sppeach!
>> Please speak now...
[You]
great I want to get a good picture
>> Waiting for response from ChatGPT...
[ChatGPT]
Absolutely! It will be stunning up there. The view from the top is going to be worth the climb
. Let's make sure we capture that moment with a great picture!
```

図 4.2 プロンプトによる制御を加えて英会話を行なった様子

```
let's have some lunch fast
>> Waiting for response from ChatGPT...
[ChatGPT]
Sure, we can have a quick lunch. Let's find a comfortable spot, it's important to keep up our
strength before the final climb.
>> Generating audio file....
>> Ready to Sppeach!
```

図 4.3 プロンプトによる制御を加えて英会話を行なった様子

制御を行っていない場合と比べて、返答が短くなり、表現も易くなった。

### 4.3 感情を入力する機能を追加した英会話

最後に、感情を入力しながら会話を行なった。話者の一つ目の会話は「悲しい」、二つ目の会話は「楽しい」、三つ目の会話は「喜び」、という感情を入力して会話を行なっている。図 4.4 は実際に会話を行なった様子である。

```
emotion:Sorrowful.
>> Please speak now...
[You]
we are almost at the top of the mountain
  >> Waiting for response from ChatGPT...
[ChatGPT]
Wow, that must be a tough climb. But think about the beautiful view we'll have at
the top. Let's keep going!
>> Generating audio file....
>> Ready to Speeach!
emotion:Fun.
>> Please speak now...
[You]
great I want to get a good picture
  >> Waiting for response from ChatGPT...
[ChatGPT]
Yes, it's a good idea! The photo from the top of the mountain will be amazing. We
can show it to our friends later!
>> Generating audio file....
>> Ready to Speeach!
emotion:Happy.
>> Please speak now...
[You]
let's have some lunch fast
  >> Waiting for response from ChatGPT...
[ChatGPT]
Sure, let's eat quickly so we can continue our adventure. What are you in the mood
for? I brought sandwiches and apples.
>> Generating audio file....
>> Ready to Speeach!
```

図 4.4 感情を入力して会話を行なった様子

感情を入力していない時に比べ、入力した感情に合わせた返答を行なっている。悲しい、怒りなどの強い感情であれば会話に及ぼす影響は大きかったが、楽しい、喜びといった感情は、会話に及ぼす影響が少なかった。

## 5章 結論

ChatGPT を用いた英会話学習に関する研究を行なった。その結果、システムを構築し、英語学習を楽しくするという目標の一部を達成することができた。

本研究ではベースとなるシステムを構築することができたがいくつか問題点が残ってしまった。まず実際の目標である、カメラで表情を読み取り、読み取った表情から感情を読み取る機能は実装できなかった。理由としては技術的に非常に難しく、先行研究も少ないため、参考となるものが少なかったためである。実際にこの機能を実装できれば、より良いシステムが構築できたと思われる。

2つ目の問題点としては、システムを実行した際に、発話者が英語を発してから、システムが英語を返すまでにラグがあることである。今回構築したシステムでは ChatGPT からの返答を一度音声データにしてスピーカーから再生しているためラグが発生してしまう。一連の動作で発生してしまうラグを減らすために、ChatGPT からの返答を音声に変換する際に、一度で変換するのではなく、生成された回答を部分的に変換し再生することでより高速に返答することができるのではないかと考えている。

1年間この研究を行い、ChatGPT や API, text-to-speech, speech-to-text 技術に関して勉強することができた。課題点もいくつか残る結果となったが、これまでに学んだ技術を生かし、これからの情報技術の発展に活かしていきたいと考える。

## 参考文献

- [1] 武井一巳, 「10 倍速で成果が出る! ChatGPT スゴ技大全」, 翔泳社, pp. 223-225, 2023 年
- [2] 谷口恵子, 「AI 英語革命」, リチェンジ, pp.239-240, 2023 年
- [3] SpeechRecognition, PyPI, <https://pypi.org/project/SpeechRecognition/>
- [4] gTTS, PyPI, <https://pypi.org/project/gTTS/>
- [5] ChatGPT API とは? 始め方や使用時の注意点、活用事例を紹介, AIsmiley, [https://aismiley.co.jp/ai\\_news/what-is-the-chatgpt-api/](https://aismiley.co.jp/ai_news/what-is-the-chatgpt-api/)

## 謝辞

本論文を作成にあたり,丁寧な御指導を賜りました蚊野浩教授に感謝いたします.



## 付録 開発したプログラムとその説明

### Speak\_English.py

ChatGPT API, SpeechRecognition, gTTS を動かすためのファイル.  
ChatGPT API で発話者の発言に対する返答を行い, SpeechRecognition で発話者の発言をテキストデータに変換, gTTS で ChatGPT からの返答を音声に変換している.以下に主に使う関数を説明する.

- `recognize_speech`  
音声認識を行う関数
- `text_to_speech`  
テキストデータを音声に変換する関数
- `play_mp3_blocking`  
`text_to_speech` で変換した音声を再生する関数
- `create_image_button_app`  
感情を入力するための UI を表示する関数