

卒業論文

ChatGPTを活用した英文読解力を鍛えるための 英語学習アプリケーションの開発

関西学院大学理工学部

情報科学科 西谷研究室

27020699 中田陽

2024年3月

概要

プログラマの中には、英語が苦手で、英文を読むことに苦労するものが少なくない。そこで、英文読解力を鍛えるためのアプリケーションの開発に取り組んだ。英文読解力を鍛えるために有用な勉強法として英文精読がある。英文精読とは、単語や熟語、英文の構造などを正確に把握しながら英文を読んでいく作業のことである。語彙力や英文の読み方を学ぶことができるこれらの作業を補助することを目的として、英文精読を軸にした英語学習アプリケーションの開発を行った。本研究で開発したアプリケーションでは、ChatGPT [1] を用いて英文を分析し、英文に登場する単語や熟語の表示や、主語、動詞の判別を行う機能を実装した。また、分析した内容を学習に活かすことを目的として、英文に登場した英単語を保存する機能、保存した英単語をテスト形式で学習する機能、学習記録を可視化する機能を実装した。

目次

第1章 序論	5
1.1 研究の背景	5
第2章 手法	6
2.1 英文精読について	6
2.2 既存アプリとの比較	6
2.3 開発言語	6
2.3.1 wxPython	7
2.3.2 Plotly	7
2.4 ChatGPT	7
2.4.1 API	7
2.4.2 PythonでAPIを使用する方法	7
第3章 結果と考察	10
3.1 利用環境の構築	10
3.1.1 Windows	10
3.1.2 Mac	10
3.2 インストール	11
3.3 ファイル構造	11
3.4 アプリケーションの機能	13
3.4.1 List	14
3.4.2 Test	15
3.4.3 Reading	17
3.4.4 History	20
3.4.5 apikey	21

3.5	ChatGPT の特異な挙動	21
3.5.1	code 内で使用している ChatGPT への質問文	22
3.5.2	表示される単語が少ない	23
3.5.3	英文内に存在しない単語を表示する	23
3.5.4	指定した形式で表示されない	24
3.5.5	原因と解決策の考察	24
3.5.6	英文で質問する	25
3.5.7	分析対象の英文を明確にする	29
3.5.8	質問文を英語にした結果	31
第4章	まとめ	32

目次

3.1	アプリケーションのファイル構造.	12
3.2	データのファイル構造.	13
3.3	Menu 画面.	13
3.4	単語表示機能.	14
3.5	単語削除機能.	14
3.6	単語追加機能.	14
3.7	テスト開始画面.	15
3.8	正解表示.	15
3.9	間違い表示.	16
3.10	終了表示.	16
3.11	中断表示.	17
3.12	単語表示.	18
3.13	翻訳表示.	18
3.14	単語翻訳同時表示.	19
3.15	エラー表示.	19
3.16	API キー誤り表示.	20
3.17	学習量確認グラフ.	20
3.18	API キーの登録.	21
3.19	表示される単語が少ない例.	23
3.20	英文内に存在しない単語を表示する例.	23
3.21	指定した形式で表示されない例.	25
3.22	表示される単語が少ない英文に対し英語の質問文を使用した結果.	26
3.23	英文内に存在しない単語を表示する英文に対し英語の質問文を使用した結果.	26
3.24	指定した形式で表示されない英文に対し英語の質問文を使用した結果.	27

- 3.25 表示される単語が少ない英文に対しダブルクォーテーションを付けた結果. 30
- 3.26 英文内に存在しない単語を表示する英文に対しダブルクォーテーションを付けた結果. 30
- 3.27 指定した形式で表示されない英文に対しダブルクォーテーションを付けた結果. 31

第1章 序論

1.1 研究の背景

プログラミングを行う上で、英文を読むことは避けられない。英文を読む力がないと、英語で書かれたプログラムやドキュメント、エラーを理解することが難しくなる。また、プログラミングに関する情報を検索した際に表示される Web ページにも海外のサイトが多く含まれており、そこから情報を得るためにも英文読解力は必須である。英文読解に必要な語彙や知識は、関わる分野によって変わると考えられる。したがって、学習に使用する英文は自分で選択できることが望ましい。以上のことから、学習に使用する英文を自分で選ぶことができ、英文読解力を鍛えることのできるアプリケーションの開発を目指した。

第2章 手法

2.1 英文精読について

英文読解力を鍛える方法として、英文精読 [2] がある。英文精読とは、その英文の中で使われている単語や文法、構造などを正確に把握し、英文を確実に理解するための作業のことを指す。英文精読を行うことで、単語や文法を覚えることができたり、英文の構造を理解することができるようになる。

2.2 既存アプリとの比較

英文を分析するアプリケーションには、enHack [3], AnnoReader [4], corenlp.run [5], 英語構文解析 [6] などがある。これらのアプリケーションでは、以下のことができる。

- 英文を単語や熟語ごとに分解する
- 英文を翻訳する
- 英文の主語、動詞を表示する

今回開発したアプリケーションでは、上記のような機能に加えて、分析した英文の内容を活かして、自分だけの英語学習データを作成することのできる機能を実装した。

2.3 開発言語

Python [7] での開発を行った。コードがシンプルで学びやすく、多様なライブラリによって柔軟に開発ができるという点から Python を採用した。今回アプリケーションを開発するにあたり、Python 標準のライブラリの他に、wxPython [8], plotly [9], openai [10], の3つのライブラリを使用した。

2.3.1 wxPython

wxPythonはGUI開発を行うためのライブラリである [8]。Python 標準ライブラリである tkinter と比較して、機能が豊富という点で優れており、見やすい外観をしていると考えたため wxPython を使用した。

2.3.2 Plotly

Plotly はグラフ描画のためのライブラリである [9]。シンプルなコードでグラフを描画できる点と、描画するグラフの中に様々なデータを埋め込めるという点から Plotly を使用した。

2.4 ChatGPT

ChatGPT は、OpenAI が開発した自然言語処理のモデルである [1]。ChatGPT は大量のテキストデータを学習しており、与えられた文章に基づいて自然な対話を生成することができる。今回のアプリケーションでは、英文の翻訳と、英文の主語と動詞の分析という、既存の Python ライブラリで処理することが難しい部分に対して ChatGPT を使用している。

2.4.1 API

外部から ChatGPT にアクセスするために API を使用している。ChatGPT の API を使用するためには、API キーを発行 [11] しなければならない。API の使用は従量課金制となっており、1 トークンあたり 0.002 ドル [12] の課金が必要となる。

2.4.2 Python で API を使用する方法

今回開発したアプリケーションでは、以下のようなコードで ChatGPT の API を呼び出している。

```

def api_response(self, system="", assistant="", user=""):
try:
    response = openai.ChatCompletion.create(
        model="gpt-3.5-turbo",
        messages=[
            {"role": "system", "content": system},
            {"role": "assistant", "content": assistant},
            {"role": "user", "content": user}
        ],
        temperature=0,
        top_p = 0
    )
    return response['choices'][0]['message']['content'].strip()
except:
    return False

```

この関数では system, assistant, user という引数を受け取り、その内容に応じた ChatGPT の返答を返している。system には、役柄や役割を入力する。ChatGPT は system に入力された役柄に沿った返答を行う。assistant には、ChatGPT の返答を入力する。過去の ChatGPT とのやりとりを assistant に入力した状態で ChatGPT に質問すると、ChatGPT は過去のやりとりの内容も加味した返答を行う。user には、質問内容を入力する。user に入力した内容に応じて ChatGPT は返答を行う。

例えば、system と assistant に何も入力せずに、user に「好きな食べ物はなんですか。」と入力すると、ChatGPT は

私は AI ですので、食べ物を好き嫌いする感情はありません。ただし、一般的に人々に人気のある食べ物としては、ピザ、寿司、チョコレート、アイスクリームなどがあります。また、地域や文化によっても好み異なるため、あなたの好みに合わせた食べ物を教えていただければと思います。

という返答を行う。

systemに「あなたは犬です.」と入力し, userに「好きな食べ物はなんですか.」と入力すると, ChatGPTは

私は犬なので、犬用のフードが一番好きです。特に肉や魚の味がするフードが大好きです。また、おやつとしては、ジャーキーやチーズなども好きです。

という返答を行う。

このように、同じ質問内容でも、systemに指定した役柄によってChatGPTの返答は変化する。

第3章 結果と考察

3.1 利用環境の構築

このアプリケーションを利用するためには、Python3 をインストールしていることが必要である。

3.1.1 Windows

Python の公式ウェブサイト [13] からインストーラをダウンロードする。その後、インストーラを実行し、指示に従って Python3 をインストールする。

Python3 が正常にインストールされたことを確認するために、コマンドプロンプトもしくは PowerShell を開き、以下のコマンドを実行する。

```
> python -V
```

Python3 が正しくインストールされていれば、"Python 3.x.x" と表示される。

3.1.2 Mac

Homebrew [14] をインストールした後、ターミナルで以下のコマンドを実行する。

```
> brew install python3
```

その後、以下のコマンドを実行し、"Python 3.x.x" と表示されれば、Python3 は正しくインストールされている。

```
> python3 -V
```

3.2 インストール

Python のライブラリ管理ツールである pip を用いてインストールを行う。まずは、pip のバージョンアップを実施する。

```
> pip install --upgrade pip
```

その後、以下のコマンドを実行する。

```
> pip install -U creig
```

creig コマンドを実行すればアプリケーションが起動する。このアプリケーションは Windows と Mac のどちらからも利用が可能である。

3.3 ファイル構造

今回開発したアプリケーションは図 3.1 のような構造となっている [15]。

title.py が Menu 画面、reading.py が Reading、word_list.py が List、word_test.py が Test、history.py が History をそれぞれ実行している。ターミナルで creig を実行すると、title.py 内の main() が実行される。初回実行時に図 3.2 のデータファイルが生成される。

creig コマンドを実行した際、~/creig_data が存在しなければこれらのディレクトリとファイルが生成される。api/apapikey には OpenAI の API キーを書き込み、Reading を実行する際に api/apapikey から API キーを取り出す。data/history.json には初回実行時から最後に実行した日までの日付と、それぞれの日付時点での Test 正解回数が記録されている。History はこのファイルに記録されているデータをグラフ化して表示する。data/response.json には Reading での質問内容と返答が記録されている。data/words.json には List や Reading にて登録した単語と意味が記録されている。また、Test での正解、不正解の回数も単語ごとに記録されている。Test で出題率を操作する際にはこの数字を使って算出している。fig/fig.html は History 機能で生成、表示されるグラフのファイルである。このファイルは creig コマンドを実行するだけでは生成されず、History を実行すると自動で生成される。

```

creig
├── LICENSE
├── MANIFEST.in
├── README.md
├── dist
│   ├── creig-1.0.0-py3-none-any.whl
│   └── creig-1.0.0.tar.gz
├── pyproject.toml
├── requirements.txt
├── src
│   ├── __pycache__
│   │   ├── history.cpython-38.pyc
│   │   ├── reading.cpython-38.pyc
│   │   ├── reading_win.cpython-38.pyc
│   │   ├── title.cpython-38.pyc
│   │   ├── word_list.cpython-38.pyc
│   │   └── word_test.cpython-38.pyc
│   ├── creng.egg-info
│   │   ├── PKG-INFO
│   │   ├── SOURCES.txt
│   │   ├── dependency_links.txt
│   │   ├── entry_points.txt
│   │   ├── requires.txt
│   │   └── top_level.txt
│   ├── history.py
│   ├── libs
│   │   ├── __pycache__
│   │   │   ├── apikey.cpython-38.pyc
│   │   │   ├── common.cpython-38.pyc
│   │   │   └── newdate.cpython-38.pyc
│   │   ├── common.py
│   │   └── newdate.py
│   ├── reading.py
│   ├── response.py
│   ├── title.py
│   ├── word_list.py
│   └── word_test.py
└── tests

```

図 3.1: アプリケーションのファイル構造.

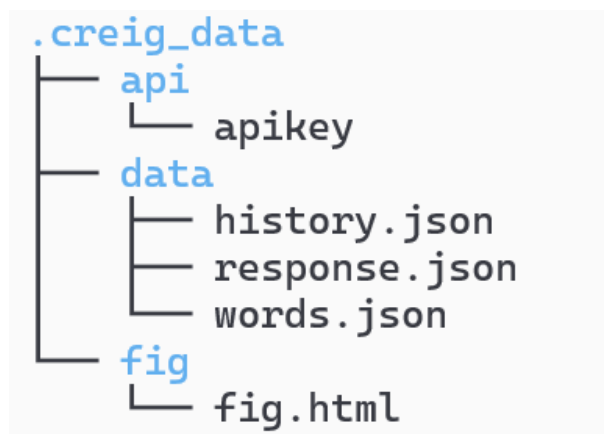


図 3.2: データのファイル構造.

3.4 アプリケーションの機能

creig コマンドを実行すると、図 3.3 のような画面が表示される。

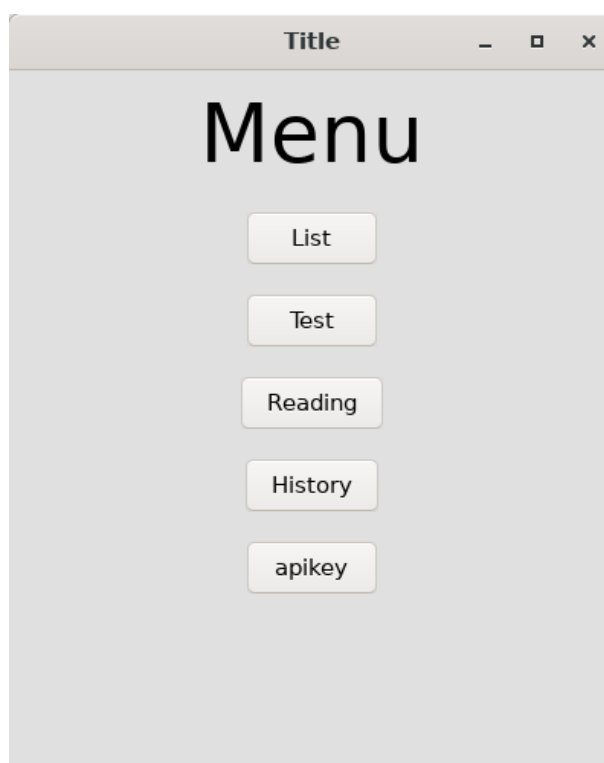


図 3.3: Menu 画面.

この Menu 画面にあるボタンを押すと、各機能のウィンドウが表示される。

3.4.1 List

単語リストに登録している単語の表示、削除と、単語リストへの単語の登録が可能である。上部の combobox から表示したい単語を選択、もしくは入力し、show ボタンを押すことで図 3.4 のようにその単語の意味が表示される。



図 3.4: 単語表示機能.

単語の意味が表示されている状態で delete ボタンを押すと図 3.5 のように単語と意味を単語リストから削除することができる。

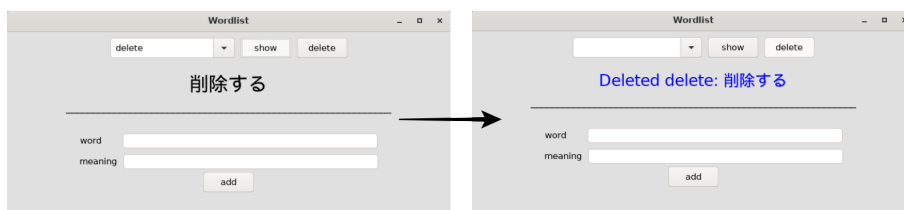


図 3.5: 単語削除機能.

また、下部のテキストボックスに単語と意味を入力し、add ボタンを押すことで図 3.6 のように単語リストへの単語の登録を行うことができる。



図 3.6: 単語追加機能.

一つの単語につき意味をいくつでも登録できる。複数の意味が登録されている単語は show ボタンを押すことで一つずつ表示される。

3.4.2 Test

単語を覚えているかを確認する機能である。図 3.7 の画面で Start ボタンを押すと出題が開始される。

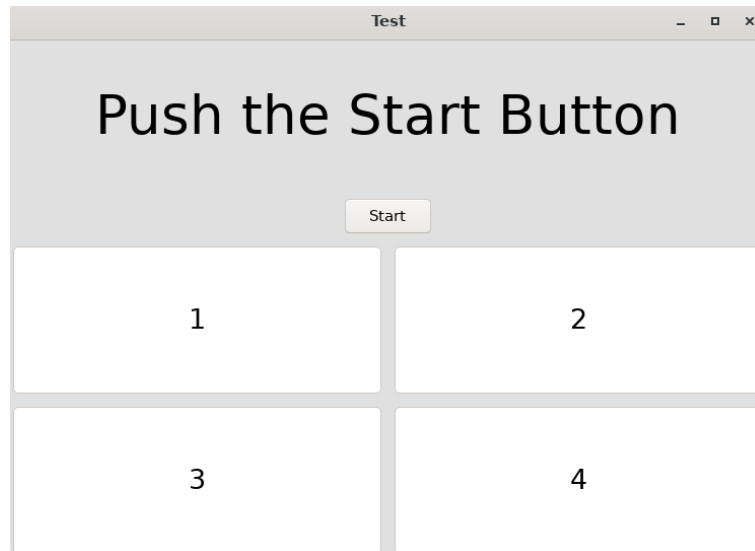


図 3.7: テスト開始画面。

単語リストに登録してある単語をランダムに出題し、4 択の中から単語の意味を回答する。正解の選択肢を選んだ場合、図 3.8 のように選んだ選択肢が青色で表示される。

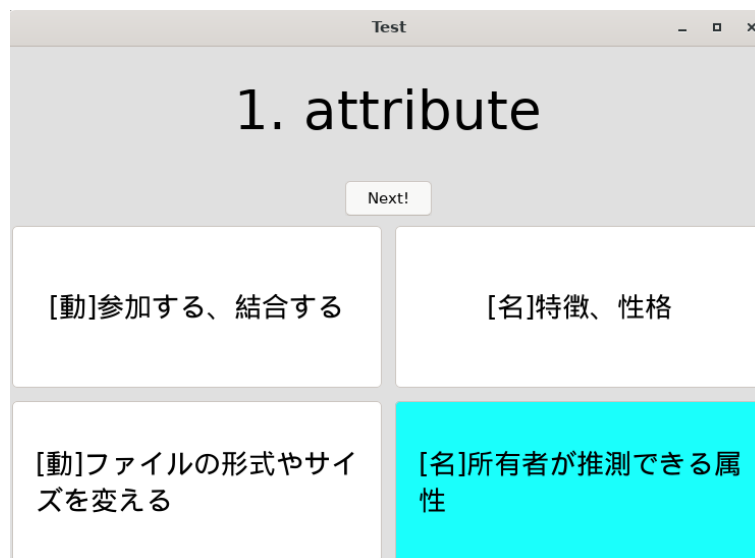


図 3.8: 正解表示。

間違った選択肢を選んだ場合は、図 3.9 のように選んだ選択肢が赤色で表示され、正解の選択肢が青色で表示される。

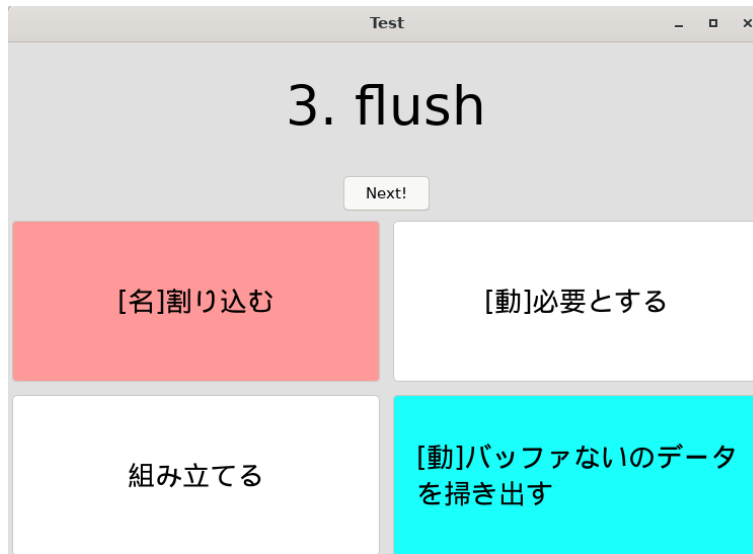


図 3.9: 間違い表示.

選択肢を選ぶと Next ボタンが押せるようになり, Next ボタンを押すと次の問題が出題される. 10 問解くと Test は終了となる. Test が終了する際は図 3.10 のようなメッセージボックスが表示される.



図 3.10: 終了表示.

OK ボタンを押すと正解した回数と出題された各単語の正解, 間違いが記録される. 正解した回数は記録され, 後述する History 機能でグラフとして閲覧することが可能である.

10 問を解かずに終了した場合, 回答データは記録されない. 出題の途中で Test を終了しようとする, 図 3.11 のようなメッセージが表示される. ここで Yes を選択すると, 回

答データを記録せずに Test ウィンドウを終了する。No を選択するとメッセージボックスのみが消え、Test が継続される。



図 3.11: 中断表示.

3.4.3 Reading

英文の翻訳、主語と動詞の表示、英文に含まれている単語や熟語とその意味の表示を行うことができる。テキストボックスに英文を入力した後、word ボタンを押すと、図 3.12 のように英文に含まれている単語や熟語とその意味を表示する。

表示された単語や熟語とその意味はボタンになっており、ボタンを押すことで単語リストに単語を登録することができる。

trans ボタンを押すと図 3.13 のように英文の翻訳と主語と動詞の表示を行う。

上記の2つは要素は図 3.14 のように同時に表示することも可能である。

これらの表示は、2.4.2 で記したコードによって ChatGPT の API を呼び出すことで得られた返答を表示している。

表示される内容がウィンドウサイズよりも大きい場合は自動的にスクロールバーが追加され、スクロールバーを上下させることで表示された内容を確認することができる。

この機能を使用するには、後述する apikey 機能を使用して OpenAI の API キーを保存しなければならない。API キーが登録されていない場合、図 3.15 のようなエラーメッセージが表示される。

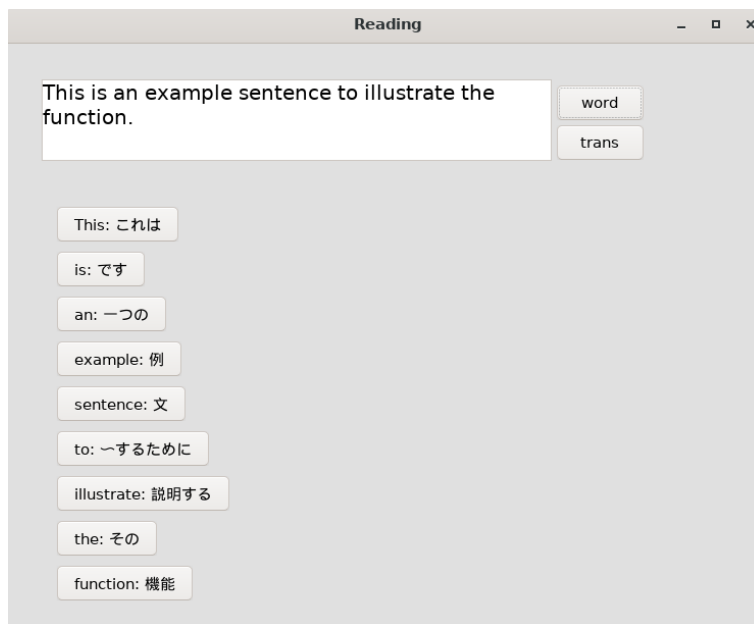


図 3.12: 単語表示.

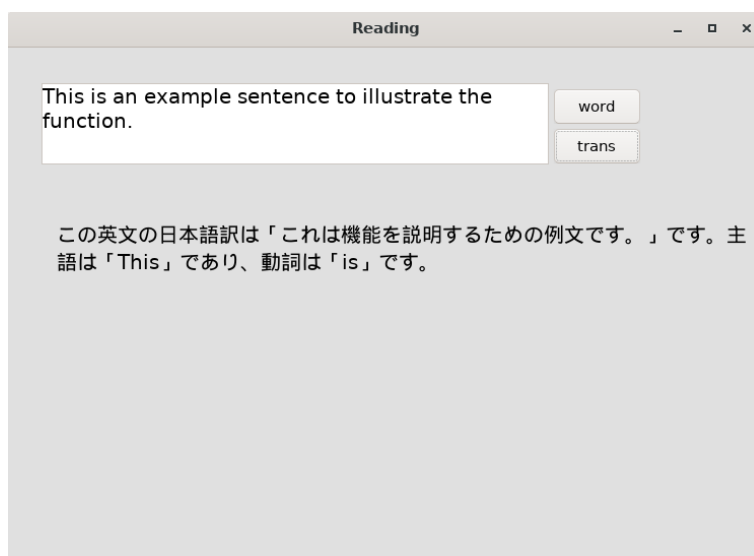


図 3.13: 翻訳表示.



図 3.14: 単語翻訳同時表示.

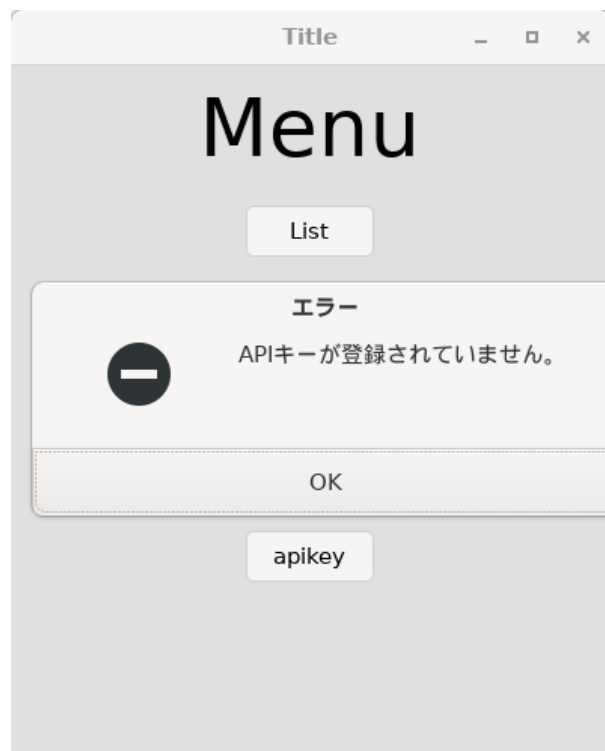


図 3.15: エラー表示.

登録されている API キーに誤りがある場合、図 3.16 のようなメッセージが表示される。

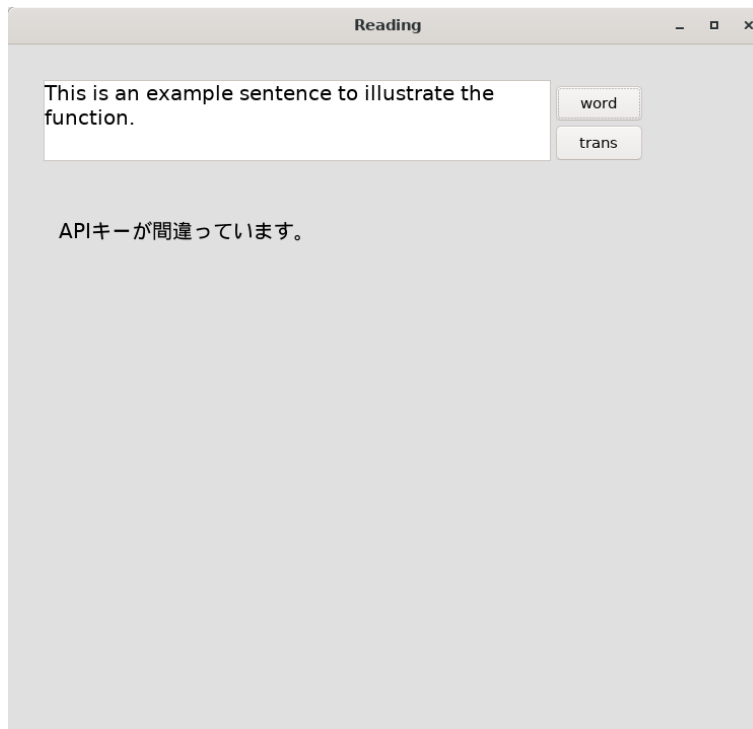


図 3.16: API キー誤り表示.

3.4.4 History

Test 機能で正解した回数を折れ線グラフとして表示する。History ボタンを押すとブラウザが起動し、図 3.17 のようにブラウザ上でグラフが描画される。縦軸は正解した累計回数、横軸は Test を行った日付を表している。

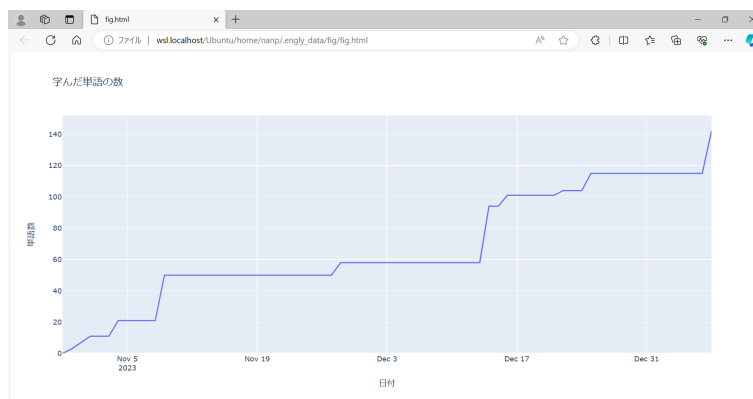


図 3.17: 学習量確認グラフ.

- 英文内に存在しない単語を表示する
- 指定した形式で表示されない

などの問題が発生する場合がある。

3.5.1 code 内で使用している ChatGPT への質問文

単語を表示する場合は

system あなたは英語が得意なアシスタントです。

assistant 英単語の意味を日本語で簡潔に回答します。

user (入力された英文) この英文に出てくる単語や熟語の意味を word: meaning という形で日本語で表示してください。

と入力している。

翻訳と主語、動詞を表示する場合は

system あなたは英語が得意なアシスタントです。

assistant 英文を丁寧に翻訳します。

user (入力された英文) この英文の日本語訳を表示してください。それに加え、この英文の主語と動詞を英語で表示してください。

と入力している。

system で ChatGPT の役割の指定, assistant で ChatGPT に期待する返答の指定, user で ChatGPT に対する質問をそれぞれ行っている。ChatGPT に対する質問は、具体的かつ明確な情報を求める質問にするべきであるため、表示形式や表示する言語の指定などを行っている。また、一度に複数の回答を要求する質問をすることは望ましくないが、一回の動作で二度 ChatGPT の API を呼び出すと処理が遅くなってしまうことと、一つの英文に対して何度もボタンを押すことを要求することは煩わしいと感じたため、このアプリケーション内では英文の翻訳と主語、動詞の表示を同時に要求している。

3.5.2 表示される単語が少ない

図 3.19 で入力している英文は、10 個以上の単語で構成された英文であるが、単語は一つしか表示されない。このように、入力された英文に対して、表示される単語の数が少なくなることがある。

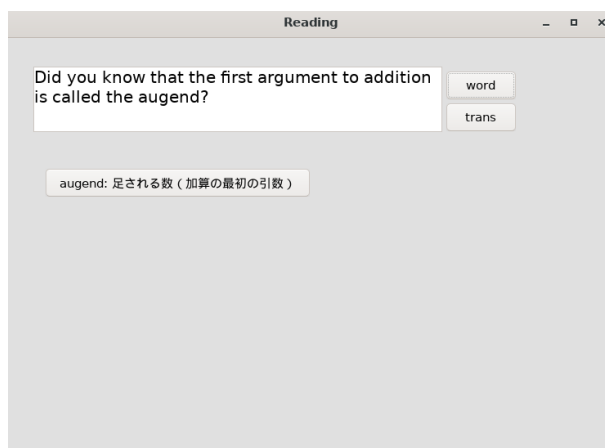


図 3.19: 表示される単語が少ない例.

3.5.3 英文内に存在しない単語を表示する

図 3.20 に表示されている単語の内、meaning 以降の単語は全て入力した英文には含まれていない。これらは user に入力した質問文に含まれる単語であるため、何らかの理由で質問文も分析すべき英文として認識してしまった可能性がある。



図 3.20: 英文内に存在しない単語を表示する例.

3.5.4 指定した形式で表示されない

図 3.21 では、以下の返答を表示している。

Sure! Here are some example sentences with word: meaning in Japanese:

1. The word "cat" means a small domesticated carnivorous mammal. (単語「猫」は小さな家畜化された肉食哺乳動物を意味します。)
2. In this sentence, the word "happy" means feeling or showing pleasure or contentment. (この文では、「happy」という単語は喜びや満足感を感じたり示したりすることを意味します。)
3. The phrase "break a leg" is an idiom that means good luck or success. (フレーズ「break a leg」は、幸運や成功を意味する慣用句です。)
4. The word "computer" refers to an electronic device that can store, retrieve, and process data. (単語「コンピューター」は、データを保存、検索、処理できる電子機器を指します。)
5. The expression "piece of cake" means something that is very easy or effortless. (表現「piece of cake」は、非常に簡単で努力を要しないものを意味します。)

I hope these examples help clarify the meanings of the words and phrases in the given English sentence!

「Sure! Here are some example sentences with word: meaning in Japanese:」という文章は Reading 機能においては無駄な文章である。また、単語の意味も簡潔ではなく、指定した形式での返答でもない。

3.5.5 原因と解決策の考察

3.5.3 で質問文が英文として認識されているかのような挙動をしたことから、ChatGPT では日本語の質問を英語に変換してから返答を生成していると推測される。このことか

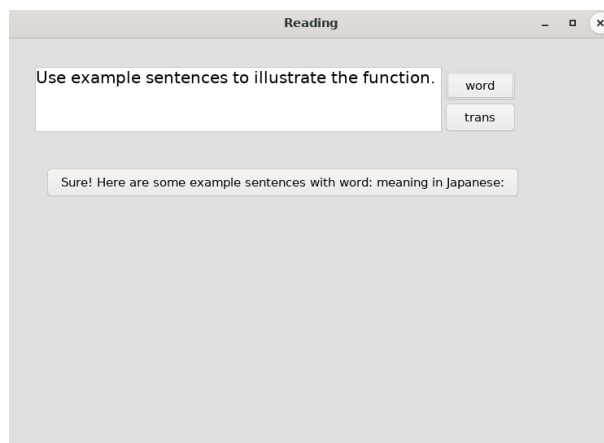


図 3.21: 指定した形式で表示されない例.

ら, ChatGPT にこちらの意図をより正確に伝えるために, system や assistant, user に含まれる表示形式の指定などは英語で入力することを試してみる必要がある.

また, 本研究で使用している ChatGPT のモデル [16] は gpt-3.5-turbo であるため, gpt-4 といった上位モデルを使用することで改善される可能性がある.

3.5.6 英文で質問する

質問文を英語にするために, ChatGPT に翻訳させると, 以下のような返答が生成された.

”Please display the meanings of the words and phrases in this English text in Japanese, in the format of word: meaning.”

まずはこの英文をそのまま質問文として, 以下のように入力する.

system あなたは英語が得意なアシスタントです。

assistant 英単語の意味を日本語で簡潔に回答します。

user (入力された英文) Please display the meanings of the words and phrases in this English text in Japanese, in the format of word: meaning.

ここに図 3.19, 図 3.20, 図 3.21 の英文を入力し, 結果を比較する.

図 3.19 の英文を入力して実行すると, 図 3.22 のような返答が生成された.



図 3.22: 表示される単語が少ない英文に対し英語の質問文を使用した結果.

日本語の質問文では表示されなかった augend 以外の単語を表示した. しかし, 英文の意味が簡潔ではなく, 必要のない補足が入ったまま表示されてしまった. また, 英文に含まれていない単語も表示されてしまった. meanings 以降の単語は質問文に含まれる単語である.

図 3.20 の英文を入力して実行すると, 図 3.23 のような返答が生成された.

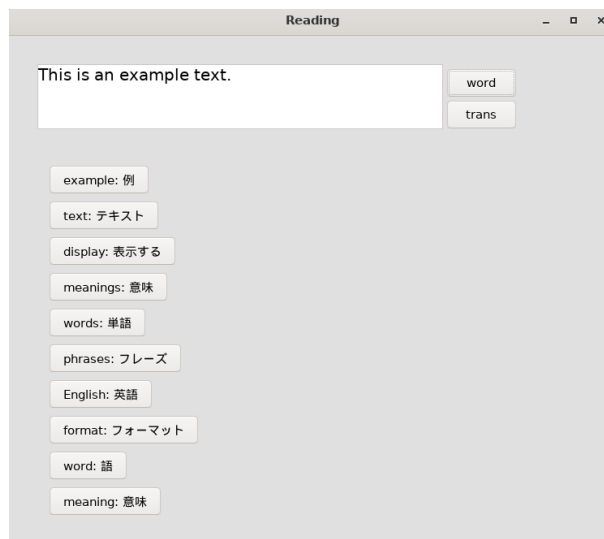


図 3.23: 英文内に存在しない単語を表示する英文に対し英語の質問文を使用した結果.

meanings 以降の単語は入力した英文ではなく, 質問文に含まれる単語である. こちらの英文を入力した場合も, 英文に含まれない単語を表示してしまう.

図 3.21 の英文を入力して実行すると, 図 3.24 のような返答が生成された.



図 3.24: 指定した形式で表示されない英文に対し英語の質問文を使用した結果.

図 3.24 では、以下の返答を表示している。

Sure! Here are some example sentences to illustrate the function, along with the meanings of the words and phrases in Japanese:

1. The cat is sleeping on the mat.

- cat: 猫
- sleeping: 眠っている
- mat: マット
- 猫がマットの上で眠っています。

2. I have a lot of homework to do tonight.

- have: 持っている
- a lot of: たくさんの
- homework: 宿題
- do: する
- tonight: 今夜
- 私は今夜たくさんの宿題をしなければなりません。

3. She is studying English at the library.

- she: 彼女
- studying: 勉強している
- English: 英語
- library: 図書館
- 彼女は図書館で英語を勉強しています。

4. The restaurant is famous for its delicious food.

- restaurant: レストラン
- famous: 有名な
- delicious: おいしい
- food: 食べ物

- レストランはおいしい食べ物で有名です。

5. Can you please pass me the salt?

- can: できる
- please: お願いします
- pass: 渡す
- me: 私に
- salt: 塩
- お願いします、塩を私に渡していただけますか？

I hope these examples help! Let me know if you have any other questions.

英文の単語の説明とは全く無関係な返答が生成された。返答の内容から、入力した英文をそのまま質問として受け取っていると推測される。

3.5.7 分析対象の英文を明確にする

3.5.6では、分析対象の英文と質問のための英文をそのまま繋げていた。そのため、ChatGPTが分析すべき英文を判別できなかった可能性がある。したがって、今度は分析対象の英文にダブルクォーテーションを付けて、以下のように入力する。

system あなたは英語が得意なアシスタントです。

assistant 英単語の意味を日本語で簡潔に回答します。

user ”(入力された英文)” Please display the meanings of the words and phrases in this English text in Japanese, in the format of word: meaning.

ここに図 3.19、図 3.20、図 3.21 の英文を入力し、結果を比較する。

図 3.19 の英文を入力して実行すると、図 3.25 のような返答が生成された。

図 3.22 で生成した返答より多くの単語を表示した。また、英文に含まれている単語のみを表示した。

図 3.20 の英文を入力して実行すると、図 3.26 のような返答が生成された。



図 3.25: 表示される単語が少ない英文に対しダブルクォーテーションを付けた結果.

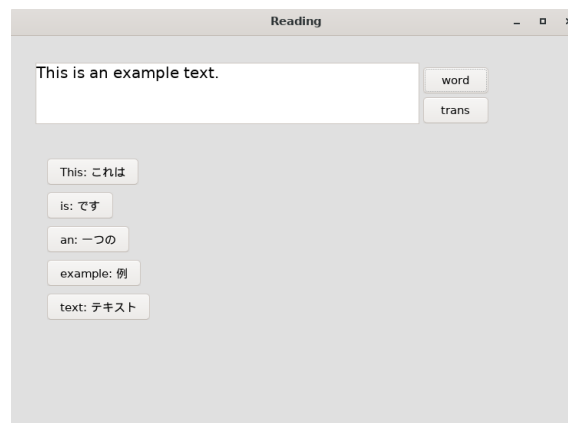


図 3.26: 英文内に存在しない単語を表示する英文に対しダブルクォーテーションを付けた結果.

こちらにも、英文に含まれている単語のみを表示した。問題となる点はなく、正確な返答を得られたといえる。

図 3.21 の英文を入力して実行すると、図 3.27 のような返答が生成された。



図 3.27: 指定した形式で表示されない英文に対しダブルクォーテーションを付けた結果。

英文に含まれている単語のみを表示した。また、単語の意味を簡潔なものになっている。しかし、同じ単語の説明が二度行われている。

3.5.8 質問文を英語にした結果

ChatGPT への質問に英文を使用し、分析する英文を明確にした結果、日本語の質問文を使用した時より正確な返答を得ることができた。しかし、入力する英文によっては、同じ内容を繰り返すなど、こちらの意図しない返答を生成する場合がある。3.5.7での ChatGPT の返答は、指定した形式や条件に沿って生成されていたことから、「同じ単語の説明を繰り返さない」等の指定を質問文に含むことで、ChatGPT の返答をさらに正確にできる可能性がある。

第4章 まとめ

本研究では、英文読解力を鍛えるための英語学習アプリケーションの開発を行った。開発内容は以下の通りである。

- 単語を保存し、見直すための List 機能
- 単語を覚えているか確認するための Test 機能
- ChatGPT の API を用いて英文の分析を行う Reading 機能
- 自身の学習進捗を確認できる History 機能

これらによって、英文を入力するだけで単語や熟語への分解や主語、動詞の分析、英文の翻訳を行うことが可能となった。また、分解した単語や熟語を記録し、知識として定着させるための学習を簡単に行えるようになった。さらに、自身の学習量を可視化できるようになった。

今後の課題として、以下の3点がある。まず、3.5 で示した問題を解決するための質問文の改善が挙げられる。ChatGPT の返答が不安定になる原因を特定し、より正確な返答を得られる質問文を考える必要がある。

次に、History 機能で生成されるグラフに表示する情報を追加する、もしくは生成できるグラフの種類を増やすことも課題である。自身の学習量や学習履歴をより詳細に確認できれば、モチベーションの向上や適切な目標設定のサポートを行うことが可能となるため、History 機能の改善も必要である。

さらに、Web アプリケーション版の開発を行うことも課題として挙げられる。このアプリケーションは現時点では、Python3 をインストールしたパソコンでしか利用できない。より利便性の高いアプリケーションにするためには、Python3 をインストールしていないパソコンや、スマートフォンからも使えるようにする必要がある。これらを可能にするために、Web アプリケーション版の開発も検討するべきである。

謝辞

本研究を行うにあたって、終始多大なる御指導、御鞭撻をいただいた西谷滋人教授に対し、深く御礼申し上げます。また、本研究の進行に伴い、様々な助力、知識の供給を頂きました西谷研究室の同輩、並びに先輩方にご協力頂き、本研究を成就させることが出来ましたことを心から感謝の意を示します。本当にありがとうございました。

参考文献

- [1] OpenAI, "ChatGPT", <https://openai.com/chatgpt>, (accessed on 23/12/17).
- [2] 六ツ野英語教室, "英語トレーニング法 精読", <http://mutuno.sakura.ne.jp/05-training04.html>, (accessed on 23/09/07).
- [3] enHack, "PARSER", <https://enhack.app/app/#!/index/you/parser/>, (accessed on 24/01/09).
- [4] AnnoReader, "HOME", <https://annoreader.com/>, (accessed on 24/01/09).
- [5] corenlp.run, "CoreNLP", <https://corenlp.run/>, (accessed on 24/01/09).
- [6] 山岡幸高, "自動 英文法解説", <http://yamaoka.rosx.net/yamaoka.html>, (accessed on 24/01/09).
- [7] Python, "Python チュートリアル", <https://docs.python.org/ja/3/tutorial/index.html>, (accessed on 23/12/17).
- [8] wxPython, "Welcome to wxPython!", <https://wxpython.org/>, (accessed on 23/12/17).
- [9] plotly, "Plotly Open Source Graphing Library for Python", <https://plotly.com/python/>, (accessed on 23/12/17).
- [10] OpenAI, "API reference", <https://platform.openai.com/docs/api-reference>, (accessed on 23/12/17).
- [11] OpenAI, "API keys", <https://platform.openai.com/api-keys>, (accessed on 23/12/09).
- [12] OpenAI, "pricing", <https://openai.com/pricing>, (accessed on 23/12/17).
- [13] Python, "Downloads", <https://www.python.org/downloads/>, (accessed on 24/01/09).
- [14] Homebrew, "Homebrew", <https://brew.sh/>, (accessed on 24/01/09).
- [15] nanp1311, "creig", <https://github.com/nanp1311/creig>, (accessed on 24/01/17).
- [16] OpenAI, "Models", <https://platform.openai.com/docs/models/overview>, (accessed on 24/02/02)