

令和3年度 卒業論文

青空文庫音読サイト
音声認識 API を用いた青空文庫の音声操作

近畿大学工学部 情報学科

情報物理研究室

1810990082 大橋雄喜

1. 研究の目的と背景

1.1. 目的

パソコンやスマホなど便利なものをより便利にすること、健常者はもちろんのこと、それだけでなくパソコンの操作が難しい腕や手などの身体障害者の方や目の不自由な方に読書をスマートスピーカー感覚で楽しんでもらうことを研究の目的とし、ボイスユーザインタフェース (VUI) に考慮してアプリケーションの開発を行う。ボイスユーザインタフェースの詳細については「2.2.1. キーワード」にて後述している。

1.2. 背景

今日では、人間の声をコンピュータに認識させることで、音声でスマホやパソコンを操作できる音声認識機能を使用することが増えてきた。Apple が提供している iPhone の音声アシスタント機能「Siri」や Amazon が開発したバーチャルアシスタント AI 技術「Amazon Alexa (アレクサ)」, Microsoft が提供しており、Windows 10 などに搭載されている音声アシスタント機能「Cortana (コルタナ)」などが例に挙げられる。音楽や天気、家電のオンオフなどスマートスピーカーは色々な機能を持っている。現在、音声認識を利用したサービスは、企業、病院、自治体でも議事録の自動作成支援や学校向けの語学学習支援などの実用システムが活用されている。

研究を行うにあたっての前段階として、ウェブアプリケーションの作成に必要な知識を深めるために、JavaScript についての学習を行った。JavaScript を学習するための書籍やウェブサイトは多く存在しており、主に書籍を利用して学習を進め、プログラムを改変して理解を深めた。その際に、代表的な Web API (Application Programming Interface) として Web Speech API や Google Books API に触れた。

私は今年度入院しており、スマホやパソコンの操作ができないほど体調が優れないことがあった。その際、音声でデバイス进行操作し、音声で読書ができればいいと感じた。また、身体障害者や目の不自由な方には、デバイスの操作が難しいことや紙の本がめくれないうなどの壁がある。そこで音声で操作することによって身体障害者や目の不自由な方にも気軽に読書をする機会を設けられるのではないかと考えた。

このような背景から、Web Speech API を使用し、青空文庫を音声で操作したいと考え、青空文庫内の書籍の音読をしようと考えたことから青空文庫音読サイトを開発することに至った。

2. 研究内容

2.1. 概要

Web Speech API を使用して青空文庫を音声で操作するアプリケーションを制作する。アプリケーションを起動してから一切のマウス、キーボード操作なしで音読させるところまでできればいいと考えた。

2.1.1. キーワード：音声認識，音声合成，VUI

● 音声認識 [1]

音声認識とは，人間の声などをコンピュータに認識させることであり，コンピュータにより音声データをテキストデータに変換する技術である。

文字列（文章）を入力する機能だけを呼び分ける場合は「音声入力」または「ディクテーション（聞き取り）」という。同音異義語が少ない欧米系の言語では 90 パーセントの認識率がある。日本語のディクテーションの認識率は 60 パーセントから 80 パーセントなので，他の言語に比べて認識率が悪い。そのため，聞き取りが完璧にできるわけではないので，目の不自由な方は実際に音声認識できているかわからない場合もある。

キーボードから文字列やショートカットを入力して，音声認識でアプリケーションを操作することも可能である。そのような音声でアプリケーションを操作することは「音声操作」という。

本研究で使用している機能は，検索バーに検索する内容を表示させるための「音声入力」，青空文庫の書籍を音読させるために音声で操作する「音声操作」である。

● 音声合成 [2]

音声合成とは，人間の声を人工的に生成することであり，合成された音声を合成音声と呼ぶ。

テキスト（文章）を入力し，望む言語内容の音声を生成することを「テキスト音声合成（Text To Speech）」と呼び，本研究では「テキスト音声合成」の機能を使用している。テキスト音声合成では，入力されたテキスト（文章）の読み方を正しく推定することが必要である。しかし，日本語の漢字の音読み・訓読み，同音異義語の区別やアクセントの推定を正しく行うには困難である。

音声合成では特定の属性を指定した合成音声を生成する 경우가多く，属性には以下のようなものがある。

- ・話速（合成音声の話す速度のこと）
- ・大きさ（音量の設定や強弱のアクセントのこと）
- ・音高（ピッチやイントネーションのこと）
- ・音色（音の質を表現するためのもの。「おんしょく」と読む。）
- ・性別

以上の属性から，本研究で用いる Web Speech API で設定できる属性は，「話速」，「音量」，「ピッチ」，「性別」である。そのため，テキスト音声合成の問題，設定しきれない属

性があるため、違和感のある漢字の読みや発音、イントネーションをしてしまうことがある。

- ボイスユーザインタフェース (VUI), UI の進化の歴史とこれからの展望

[3][4][5][6][7]

ボイスユーザインタフェースは、コンピュータと人間のやり取りを音声によって行う音声操作やシステムからの応答を合成音声によって伝達する方式である。ボイスユーザインタフェースの処理の手順とエラーが起こりやすい箇所を示す。

手順 1. 音声から文字への変換

ユーザの発話を音声認識によりテキストに変換する。現在の音声認識精度（日本語では 60%~80%程度の認識精度）では、ユーザの発話した内容と異なるテキストに変換される場合がある。

手順 2. 発話と該当する処理をつなぎ合わせる

ユーザの発話に対して対応する特定の処理を行う必要がある。青空文庫音読サイトの例では、「検索」と発話することで書籍の検索を行う処理が実行されることである。

手順 3. テキストから合成音声により伝達する

検索処理により HTML に出力したテキストを合成音声によりユーザに伝達することができる。現状の音声合成精度（日本語、特に漢字では 5 割程度の精度に感じた。）では、誤った発音で伝達される場合がある。

このようなシステムは、下の図に示すように、音声認識・発話理解・音声合成の 3 つの要素とバックエンドアプリケーションから構成される。このような処理を逐次的処理と呼ぶ。

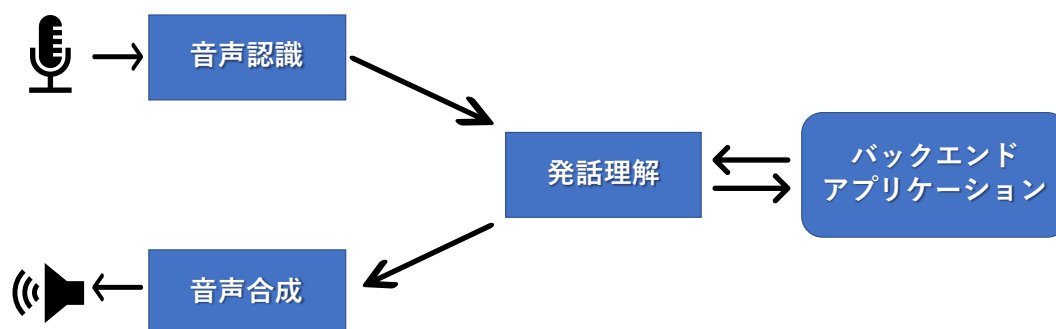


図 1 逐次的処理に基づく VUI システムの構成

また、ユーザインタフェースの進化の歴史や今後の展望についてである。次のように示した。

① CUI (Character User Interface)

文字列によるやり取りを行うインタフェースで、キーボードなどから文字列（コマンド）を入力し、ウィンドウに文字列を出力する。CUIはCLI (Command Line Interface)とも呼ぶ。

② GUI (Graphical User Interface)

GUIとは、現在最も代表的なインタフェースであり、情報やアイコンをグラフィック要素に置き換えた表示体系で、視覚的にコンピュータを操作する体系である。

③ NUI (Natural User Interface)

NUIとは現在社会に浸透しているインタフェースであり、GUIよりもさらに直感的な操作が可能になり、人間が日常的に行っている動作の延長でコンピュータの操作を可能にするユーザインタフェースである。VUIはNUIの一種であり、非接触UIもNUIに含まれる場合がある。非接触UIとはカメラやセンサで指や手での動作を検知し、操作する機器に触れないユーザインタフェースであり、コロナ禍で注目が集まるユーザインタフェースである。

④ OUI (Organic User Interface)

NUIの先にはOUIが登場してくる可能性がある。OUIとは物理的な入力によって変化する、平面ではない装置を持っているユーザインタフェースである。SFの世界のような機器が登場される可能性があり、カナダでは紙のようなスマートフォンが開発された。

ここからは完全に個人の意見となるが、近未来のユーザインタフェースでは、チップユーザインタフェースが登場すると思う。チップユーザインタフェースとは、人間の手の甲にマイクロチップを埋め込み、スマホなどと連携して、マイナンバーなどの個人情報やSuicaなどの交通系ICの情報を付加することでチップによる公共の情報機器を操作することが可能になると考える。現にスウェーデンでは数千人が体内にチップを埋め込んでいる。また、ペットショップやブリーダー等で販売される犬や猫にはマイクロチップの挿入が義務化されている。このことから、チップユーザインタフェースが利用可能になると考える。

2.1.2. 全体フロー

アプリケーションを使用する全体フローは大きく分けて3つのステップに分けられる（音読を開始するまでの手順）。

- ① 自サイト内の google 検索バーで「青空文庫 作品名」で検索，音声入力の場合はスペースキーを入力すると音声認識が開始されるので，「作品名」のみで検索する．音声入力の場合は始めから「青空文庫 」が入力された状態なので「作品名」のみの検索で，書籍の検索が可能である．
→音声認識が開始してから3秒経過した時点で，自動で音声認識を終了し，検索する．
- ② トップの検索結果を自動で選択し HTML 上に表示する．
- ③ 音声合成の開始ボタンをクリックすると，音声合成（書籍の音読）が開始される．
- ④ 音声入力していない場合であれば，シフトキーを入力して合言葉を発声すればいつでも合言葉機能が使用可能である．

本研究で制作したアプリケーション（青空文庫音読サイト）は下の図である．
以下，本アプリケーションのアドレスである．

<https://buturi.heteml.net/student/2021/ohashi/subject/kadai02/WebSpeechAPI/index.html>



図 2 青空文庫音読サイト

2.1.3. 機能

- 音声認識で青空文庫内の書籍の検索

スペースキーを入力すると音声認識ボタンがクリックされ，音声認識を開始する．また，マウス操作ができる場合は，音声認識ボタンを設置しているので，音声認識ボタンをクリックすることで音声認識を開始することができる．音声認識が開始されると自動的に

「青空文庫 (空白)」が検索バーの中に記述され、書籍の題名を発声することで検索される。

音声認識を開始してから3秒後のタイミングで音声認識を中止し、「青空文庫 書籍の題名」で検索する。検索結果の一番上の書籍を自動的に選択し、書籍の内容をHTML内に記述する。

- 検索した本の内容の読み上げ

検索された書籍はタグを含むJSON形式のオブジェクトで出力されるため、文字列に変換し、タグを削除している。書籍の内容が文字列に変換されHTMLに記述されているので、書籍の内容に音声合成を行うことで、書籍の読み上げが可能となる。なお、タグを削除して文字列で表示する方法は、正規表現を用いる。正規表現については「使用技術 2.2.8. 正規表現」にて後述している。

- 指定された合言葉での青空文庫の操作

if文を用いて合言葉や単語を登録し、音声認識で読み取った文章や単語と一致した場合に指定の処理を行うプログラムを実行する。合言葉を読み取るための音声認識は、シフトキーを入力することで音声認識を開始し、合言葉を発声することで、機能の実行が可能になる。

下記に記述したプログラムで合言葉機能を実現している。

```
if(aikotoba === "単語") function 名();  
function function 名() {  
    実現する機能  
}
```

主な合言葉と機能は下記に記述した。

- ・ 検索 (音声認識を開始する合言葉で音声認識を開始し、作品名で検索する.)
→この機能に関してはスペースキーで対応している。
- ・ 戻る, 次へ (ひとつ前のページに戻る, 次のページに進む.)
- ・ リセット (書籍音読サイトのトップページに遷移する.)
- ・ 再生, 停止 (書籍の音読の再生, 停止を実行する.)

- カメラで読書している人の表情を見て、寝ていると判断した場合は再生をストップする

→この機能については、TensorFlow を利用することで実装できるのではないかと考えていたが、時間的制約により実装ができなかった。

2.2. 使用技術について

Microsoft の提供している無料のソースコードエディタである Visual Studio Code を使用し、以下の言語等をコーディングした。また、動作環境は Google Chrome を利用する。サーバーに関してはウェブブラウザで利用できる FTP(File Transfer Protocol)システムである「ヘテムル FTP」を利用する。

2.2.1. HTML [8]

HTML とは Hypertext Markup Language の略であり、ウェブのもっとも基本的な構成要素で、ハイパーテキストを記述するためのマークアップ言語の 1 つである。HTML は Web ページの基本レイアウトに従って、ウェブページのコンテンツを記述し定義するものである。HTML 要素をタグで囲むと、単語や画像をどこかにハイパーリンクさせたり、単語を斜体にしたり、フォントを大きくしたり小さくしたりすることができる。

「ハイパーテキスト」はウェブサイト内でもウェブサイト間でも、ウェブページから別のページに接続するリンクを示す。

HTML ではウェブブラウザのテキスト、画像、その他のコンテンツを記述するために「マークアップ」を使用する。

2.2.2. CSS [9]

CSS とは Cascading Style Sheets の略であり、ウェブページのスタイルを設定するスタイルシート言語で、HTML や XML で記述された文書の体裁や見栄えを表現するために用いられる。CSS は、要素が画面上でどのように表現されるのかを定義する。

ウェブサイトは HTML と CSS から構成される。本研究の青空文庫音読サイトでも、大半のデザインは CSS で制作されている。

2.2.3. JavaScript [10][11]

JavaScript は、クロスプラットフォームで動作するオブジェクト指向のスクリプト言語で、ウェブページをインタラクティブにするために使用される。JavaScript を使用すると、静止したデータである HTML や CSS を、その場でリアルタイムに書き換えて、一部のコンテンツを入れ替えたり、画像のスライドショーのような動きを付けたりできる。

また、JavaScript 言語のコアをベースに多種多様なツールが作成され、最小限の労力で膨大な機能が利用できる。例えば、ウェブカメラからの動画ストリームの収集や操作、3D グラフィックやオーディオサンプルの生成などの機能を提供するブラウザのアプリケーションプログラミングインタフェース (API) などが利用できる。

JavaScript には、Array, Date, Math といったオブジェクトからなる基本的なライブラリや演算子、制御構造、文といったプログラミング言語の要素からなる主要な機能が含まれている。追加のオブジェクトを補うことで、マウスクリックやフォームへの入力などのユーザーイベントへの応答といったクライアントサイドの機能、アプリケーションのデータベースとデータのやり取りやサーバー上のファイルの操作といったサーバーサイドの機能を拡張できる。

本研究では、音声認識と音声認識を行う Web Speech API（詳細については「2.2.5. Web Speech API」にて後述している。）や Google の検索バーを作成し、検索ができる Custom Search API（詳細については「2.2.5. Custom Search API」にて後述している。）などの Web API で利用している。

2.2.4. PHP [12][13]

PHP は、「The PHP Group」によってコミュニティベースで開発されているオープンソースの汎用プログラミング言語およびその公式の処理系であり、特にサーバーサイドで動的なウェブページを作成するための機能を多く備えていることを特徴とする。名称の PHP は再帰的頭文字として、Hypertext Preprocessor を意味するとされており、「PHP は HTML のプリプロセッサである」と PHP 自身を再帰的に説明している。

本研究では、青空文庫の内容を取得する際に、CORS（オリジン間リソース共有）エラーを回避するために PHP を使用している。CORS の説明を下記のように記す。

- CORS（オリジン間リソース共有）

CORS は、追加の HTTP ヘッダーを使用して、あるオリジンを動作しているウェブアプリケーションに、異なるあり人にある選択されたリソースへのアクセス権を与えるようブラウザにしじするための仕組みである。ウェブアプリケーションは、自分とは異なるオリジン（ドメイン、プロトコル、ポート番号）にあるリソースをリクエストするとき、オリジン間 HTTP リクエストを実行する。

- CORS エラーを回避する仕組み



図 3 CORS エラーを回避する仕組み

sampleCS.html は自サイト (<https://buturi.heteml.net/student/2021/ohashi/>) であり、図には入りきらなかったため sampleCS.html と記述している。

自サイトと青空文庫のサイトの間に PHP を挟むことによって、CORS エラーを回避している。

下の図は CORS エラーのエラーメッセージである。これは、青空文庫のサイトにリクエストを送った時のコンソールのエラーメッセージである。Access-Control-Allow-Origin とは、どのオリジンからアクセスできるのかを示しており、今回の場合では No 'Access-Control-Allow-Origin' となっているためこのオリジンからはアクセスできないことがわかる。

```
sampleCS.html:1 Access to XMLHttpRequest
at 'https://www.aozora.gr.jp/cards/000148/files/773_14560.html?_=1635068601562'
from origin 'http://localhost:4249' has been blocked by CORS policy:
No 'Access-Control-Allow-Origin' header is present on the requested resource.
```

図 4 CORS エラーメッセージ

2.2.5. Web Speech API [14]

Web Speech API は、音声データを用いて音声認識と音声合成をウェブアプリケーションに組み込むことができる JavaScript API である。

音声認識と音声合成の概要と主な機能を下記に示す。

- 音声認識 API (Speech Recognition API)

音声認識では、デバイスからの音声認識サービスを経由した音声入力から音声の文脈を認識し、テキストデータをウェブブラウザに表示するなど適切に応答する機能を提供する。

本研究では、スペースキーを押すことで音声認識ボタンがクリックされ音声認識を開始し、書籍名を発声することで発声内容を認識し、Custom Search API で Google 検索をするということに使用している。

- 音声合成 API (Speech Synthesis API)

音声合成では、アプリケーション内のテキストを音声に合成し、デバイスのスピーカーや音声出力接続から再生する機能を提供する。合成されるテキストや発話に使用される音声などを表現するための多くのインタフェースが存在する。

本研究では、再生ボタンをクリックすることで書籍の内容を音読する機能に使用している。

2.2.6. Custom Search API

Custom Search API を使用すると、プログラムで検索結果を取得し、表示するウェブサイトやアプリケーションを開発できる。この API を使用すると、ウェブ検索もしくは画像検索のいずれかの結果を、JSON 形式で取得するリクエストを使用することができる。ただし、Custom Search API の無料枠の利用制限は、1 日に 100 リクエストである。

Custom Search API を使用するための前提条件を下記に記す。

● 検索エンジン ID

API を使用するためにコントロールパネルにて、ユーザの ID をひとつ作成する必要がある。検索エンジン ID が記載されている場所は、コントロールパネルの「設定 > 基本」のセクションにある。詳細は下記の図に記す。

新しい検索エンジン

▼ 検索エンジンの編集

青空

設定

デザイン

検索機能

統計情報とログ

ヘルプ

ヘルプ フォーラムを表示 (質問を入力)

フィードバックを送信

基本 広告 ユーザー 詳細設定

検索エンジンの基本的な情報と設定を指定します。 [詳細](#)

検索エンジン名

青空

検索エンジンの説明

検索エンジンの説明です。

検索エンジンのキーワード

検索エンジンのキーワード (例: 気候、地球温暖化、温室効果ガス)

エディション

標準

検索エンジン ID

3c473b213e0e4c131

公開 URL

<https://cse.google.com/cse?cx=3c473b213e0e4c131>

画像検索

セーフサーチ

地域

言語

日本語

図 5 検索エンジン ID

● API キー

Custom Search API は API キーを使用する必要がある。カスタム検索エンジンのユーザは、Google Cloud Platform よりキーの取得ができる。API キーが記載されている場所は、Google Cloud Platform の「API とサービス > 認証情報」のセクションにある。詳細は下の図に記す。

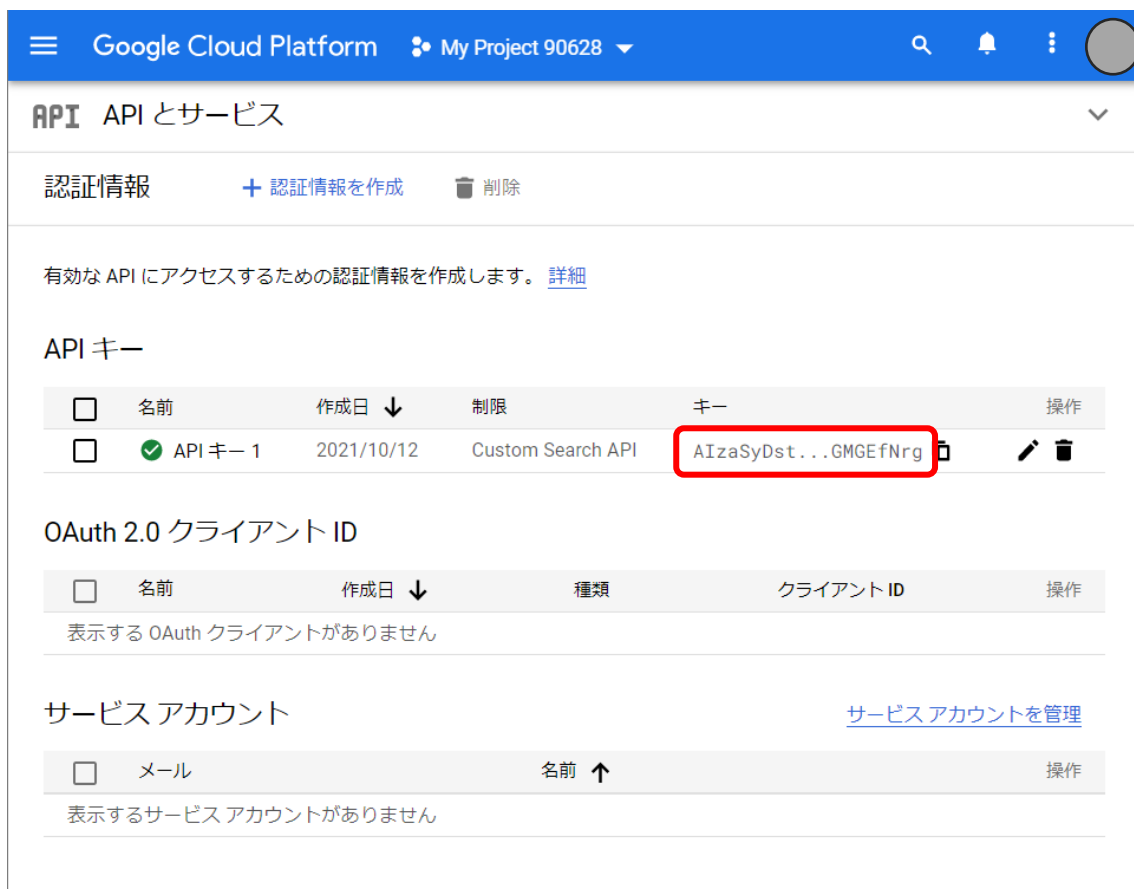


図 6 API キー

2.2.7. jQuery [15][16]

JavaScript ライブラリの jQuery は、ウェブブラウザ用の JavaScript コードをより容易に記述できるようにするために設計された。jQuery 使用される「\$」は jQuery の別名であり、エイリアスと呼ばれる。\$(function());で\$の後にすぐ関数を記述でき、\$()メソッドで記述された関数はコールバックされ、HTML がロードされるとすぐに実行できる。また、jQuery では、HTML 要素や CSS セレクタを取得し操作することもできる。特に HTML を操作することを DOM 操作と呼び、jQuery で最も大切な機能である。

そのほかの機能として、様々なブラウザで動作する使いやすい API によって、イベントハンドリング、外部のファイルを動的に読み込むことができる Ajaxなどをよりシンプルにすることができる。

本研究では、スライダーをプラグインする Slick に利用している。Slick の詳細については、「2.2.9. Slick」にて後述している。

2.2.8. 正規表現 [17]

正規表現とは、文字列内で文字の組み合わせを照合するために用いられるパターンである。JavaScript では正規表現はオブジェクトである。正規表現を使ったパターンは、文字列内で一致するものをひとつ検索し、一致した部分文字列を置換するための部分文字列で置換する `replace()` メソッドなど 8 個のメソッドを使用できる。本研究ではタグを消去するために `replace()` メソッドを使用するので取り上げた。

2.2.9. Slick

Slick (スリック) は jQuery ベースの、スライダーを作成するためのプラグインである。文字や画像が切り替わるスライダーを簡単に作成することができる。レスポンスにも対応しているため、PC だけでなくスマートフォンやタブレットで見た場合の設定が可能である。

本研究では、「よく読まれる本」を表示し、それをスライドさせることで、青空文庫でよく読まれている本をクリックすることで、書籍を読むことができる機能を導入するために利用している。

3. 青空文庫音読サイトの開発過程

3.1. アプリの雛形

卒業研究ミーティングによる経緯 (～8 月末)

身体障害者や目が不自由な人でも楽に利用でき、書籍の音読するアプリケーションを開発するという事で音声認識 API を用いてアプリケーションを制作すること決定した。音声認識 API と青空文庫を用いたアプリケーションということからアプリケーションの名前を「青空文庫音読サイト」と名付けた。

「青空文庫音読サイト」と同様のサイト「青空朗読 [18]」が存在する。青空朗読の特徴や違いなどを以下に記述した。

- ・プロのアナウンサーによる朗読であるため、聞き取りやすい。
→目の不自由な方のためのスクリーンリーダー対応版もあるが、音声読み上げソフトを使用しないと、青空朗読を利用できない。
- ・ウェブサイト内での音声操作はできない。
- ・ひとつずつ朗読するため、作品数が少ない。
- ・青空文庫の文章がなく音声のみでの朗読のため、目が見える人は物足りなく感じてしまう。
- ・作品がジャンルや時間でも検索できるので読みたい本がすぐ見つかる。
→しかし、スクリーンリーダー対応版は検索が難しい。

・音読と朗読の違い

→音読は書いてある文章を声に出して読むことであり、朗読は書いてある文章を場面の雰囲気や、作者の意図、登場人物の気持ちや性格を表現して声に出して読むことである。すなわち、合成音声は機械であり感情や意図などを考慮していないため、音声合成で文章を読むことは音読である。

3.2. 詳細設計

卒業研究ミーティングによる経緯（～9月末）

検索バーの設定を行った。Custom Search API を利用し、Google 検索をかけられるようにし、青空文庫のサイトのみがヒットするように Custom Search API を設定した。しかし、青空文庫のサイトのみでのヒットでも作品名だけでは、書籍の内容がトップの検索結果に出てこなかった。書籍の図書カードが存在しているからである。書籍の図書カードはすべての書籍に作られており、書籍の HTML より前のオリジンにコードが書かれているため、図書カードが先にヒットしてしまうと考えた。その改善策として、「青空文庫 作品名」で検索すると、書籍の内容がヒットすると明らかになった。

例. https://www.aozora.gr.jp/cards/000081/files/45630_23908.html

cards の部分に書籍の図書カードのコードが書かれており、file の部分に書籍の内容が書かれている。図書カードと作品の内容の例として使用した作品の図は下記に記載した。

図書カード : No.45630

作品名 : 〔雨ニモマケズ〕
作品名読み : 〔あめニモマケズ〕
著者名 : [宮沢 賢治](#)

[\[ファイルのダウンロード\]](#) | [いますぐXHTML版で読む](#)

作品データ

分類 : NDC 911 916
作品について :  [Wikipedia](#) 「[雨ニモマケズ](#)」
文字遣い種別 : 新字旧仮名

図 7 青空文庫の図書カード

〔雨ニモマケズ〕

宮澤賢治

雨ニモマケズ
風ニモマケズ
雪ニモ夏ノ暑サニモマケヌ
丈夫ナカラダヲモチ
慾ハナク
決シテ瞋ラズ
イツモシツカニワラッテ珥ル
一日ニ玄米四合ト
味噌ト少シノ野菜ヲタベ

図 8 青空文庫の本文

3.3. 詳細設計

卒業研究ミーティングによる経緯（～11 月末）

書籍の内容を HTML に表示させることを本アプリケーションに実装した。ただ HTML に表示させればいいだけではなく、音声合成もできるようにすることが絶対条件である。取り組んだ過程で説明する。

① `iframe`（アイフレーム）で青空文庫のサイトを表示する。

`iframe` とはインラインフレームのことであり、HTML の要素（タグ）のひとつで、ウェブページ内に矩形の領域を設け、別のウェブページなどを読み込んで表示するものである。`iframe` で表示しようとするとき問題が発生する。その問題は同一オリジンポリシーである。

同一オリジンポリシーとは、他のオリジンにあるリソースへのアクセスを制限するセキュリティのしくみである。オリジンとはドメインとプロトコルとポート番号を合わせたものである。

本研究では、自サイトの中に `iframe` で青空文庫のサイトを埋め込んでいるので、`iframe` 内の青空文庫のサイトを JavaScript で操作を実行するのが難しい。そのため、改善策の候

補として、「スクレーピング」や青空文庫のサイトにある著作権切れの書籍のファイルをダウンロードし、自サイトのみで実現する方法を考えた。

また、スクレーピングとは、ウェブサイトから必要な部分だけを抽出して変換することである。この技術を使用することで、別のサイト（今回ならば青空文庫のサイト）の必要な情報だけを抜き取れるため、同一オリジンポリシーに触れずに、JavaScript を実行できると考えた。

② Custom Search API を利用して Google 検索を行い、青空文庫の内容を表示する。

Custom Search API とは、自サイトに検索バーを設置し、プログラムで検索結果を取得し、HTML に表示することができる API である。

しかしこの方法では、検索結果を自サイトに表示できるが、検索結果をクリックすると、青空文庫のサイトに遷移してしまう。青空文庫のサイトにはプログラムの改変ができないため、音声合成を使用して音読をさせることはできない。

下の図は②の方法を使用した時の開発途中段階の青空文庫音読サイトである。



図 9 Custom Search API の途中開発画面



図 10 Custom Search API の開発途中結果画面

③ 青空文庫音読サイトから直接青空文庫にリクエストを送る。

Custom Search API を使用すると、データを JSON 形式で受け取ることが可能である。jQuery.get (\$\$.get と表記してもよい) のプログラムで青空文庫のデータを取得することができる。しかしこの方法では、CORS エラーが発生する。CORS とは、ブラウザのポリシーで異なるオリジンのアクセスを許可するものである。CORS エラーとは、同一オリジンポリシーと同じ意味であり、①の方法で音声合成ができなかった理由と同じである。

sampleCS.html から青空文庫へのウェブページにリクエストを送った場合のエラーメッセージは下の図のようになる。

```
sampleCS.html:1 Access to XMLHttpRequest
at 'https://www.aozora.gr.jp/cards/000148/files/773_14560.html?_1635068601562'
from origin 'http://localhost:4249' has been blocked by CORS policy:
No 'Access-Control-Allow-Origin' header is present on the requested resource.
```

図 11 CORS エラーメッセージ

④ PHP を経由して書籍の内容を表示する。

HTML のフォームから別のウェブサイトのデータを受け取る方法を用いて、Custom Search API で青空文庫のサイトを Google 検索した結果の第一候補のアドレス (URL) の HTML ファイルを取得して表示する。③の方法では同一オリジンポリシーに引っかかるので、PHP ファイルを作成し PHP で青空文庫のサイトにリクエストを送り、データを取得することができる。

しかし、青空文庫は承認されたウェブサイトからのリクエストしか許可していない、または、青空文庫のサーバーはプログラムからのアクセスをブロックしているため、自サイトから送るリクエストはエラーが発生する。その改善策として、アクセスが許可されているウェブブラウザ（本研究では Mozilla のサイトを模倣している。）のリクエストを模倣して、リクエストの許可やデータの取得を可能にしている。

⑤ 正規表現を使ってタグの消去と文字列処理

正規表現は、文字列内の文字の組み合わせの照合を行うので、HTML のタグを照合し `replace()` メソッドで ' ' (ヌル) に置換して、HTML のタグを消去する。タグをすべて消去すると、改行がなくなるため、書籍としては読みにくくなる。

しかし、JavaScript では正規表現はオブジェクトであり、音声合成は文字列でなくては動かないため、`textContent` で文字列に変換してから、音声合成している。

3.4. 詳細設計

卒業研究ミーティングによる経緯（～1月末）

青空文庫のサイトを音声で音声操作するための機能を実装した。実際に実装できた機能は、音声入力での音声検索機能、合言葉機能である。目標は青空文庫音読アプリを起動してから音声認識が開始し、合言葉を発声することで音声検索機能も使用可能である。また、音声認識は青空文庫のページが閉じるまで継続するつもりであった。

しかし、結果的には音声入力での音声検索機能はスペースキー、合言葉機能はシフトキーを入力することで機能を実現している。キーボード入力を分けることで、音声検索機能は認識終了後に検索するが、合言葉機能は検索を行わずに合言葉の指示に従うことができる。詳細については下記にて説明している。

● 音声入力での音声検索

スペースキーを入力することで音声認識が開始する。また、マイクのアイコンをクリックすることでも音声認識が開始する。音声認識が開始すると、検索バーの中に「青空文庫」が自動的に入力されるため、発声するのは「作品名」のみで認識可能である。

● 合言葉機能

合言葉機能の場合はシフトキーを入力することで音声認識が開始される。合言葉機能も同様に検索バーの中に「青空文庫」と自動的に入力される。そのため、合言葉も「青空文庫 合言葉」で登録しているため、発声するのは「合言葉」のみで認識可能であり、その時点で機能を実行する。

4. まとめと課題

本研究では、「健常者だけでなく、目の不自由な方、身体障害者の方にも読書をスマートスピーカー感覚で楽しんでもらうことを目的としたアプリケーション開発」を目的として、アプリを開発した。アプリを開発していく中で、実装できた部分もあったが、実装できない部分が圧倒的に多かった。特に課題として一番大きいものは、コロナ禍であることから第一の目的としていた目の不自由な方や身体障害者の方にアプリのテストやアンケートをしてもらうことができなかったことである。

その他の課題は下記に記述した。

- ・書籍の（ふりがな）を消去できずに、音声合成で同じ部分が2回音読されること。
- ・スライダーの書籍をクリックすると、HTMLに表示してほしいが、青空文庫のサイトに遷移してしまうこと。
- ・音声認識のみで青空文庫音読サイトを操作できないこと。（キーボードを使用していること）
- ・現在音読されているところにスクロールする機能が作れていない。

列挙した以外にも、実装したいと考えていた機能、修正すべき点が多々あり、完成とまではいかなかった。このような課題が多く出てしまった原因としては、プログラミング言語に対する知識不足、経験不足による開発期間の延長によるものである。開発予定期間よりも多くの時間を使ってしまうことが幾度とあり、スケジュール管理、アプリケーション開発の困難さを身にしみて感じた。一人で行うアプリケーション開発であるからこそ、スケジュール管理がすごく大事で徹底すべきであった。

本アプリケーション開発では、要件定義から実際のプログラムを書くところまで、アプリケーション開発の全体的な部分に関わり、要件定義の難しさや思い描いている完成図を作り上げていくことの難しさなど多くのことを体験することができた。

5. 参考文献・参考サイト

[1] 音声認識 | Wikipedia

<https://ja.wikipedia.org/wiki/%E9%9F%B3%E5%A3%B0%E8%AA%8D%E8%AD%98#%E6%A6%82%E8%AA%AC>

[2] 音声合成 | Wikipedia

<https://ja.wikipedia.org/wiki/%E9%9F%B3%E5%A3%B0%E5%90%88%E6%88%90>

[3] 呉越 思瑤, 酒井 哲也 | 「音声ユーザインタフェースにおける処理エラーによるユーザ frustra ションに関する調査 (2019)」

<https://db-event.jpn.org/deim2019/post/papers/218.pdf>

[4] 荒木 雅弘 | 「フリーソフトでつくる音声認識システム (第2版) -パターン認識・機械情報の初歩から対話システムまで- (2017)」, 森北出版

- [5] 諏訪 悠紀 | Voice User Interface の歴史と未来 – 人類は VUI にたどり着き、どこへ向かうのか
<https://dev.classmethod.jp/articles/history-and-future-of-vui/>
- [6] UI の歴史と将来の UI の形についてまとめてみたよ | 文系エンジニア大学生の技術ブログ
<https://www.for-engineer.life/entry/ui-history/>
- [7] 環境省_犬と猫のマイクロチップ情報登録に関する Q&A[動物の愛護と適切な環境]
<https://www.env.go.jp/nature/dobutsu/aigo/pickup/chip.html>
- [8] HTML | MDN – Mozilla
<https://developer.mozilla.org/ja/docs/Web/HTML>
- [9] CSS | MDN – Mozilla
<https://developer.mozilla.org/ja/docs/Web/CSS>
- [10] 入門編 - JavaScript | MDN – Mozilla
https://developer.mozilla.org/ja/docs/Web/JavaScript/About_JavaScript
- [11] JavaScript の基本 | MDN – Mozilla
https://developer.mozilla.org/ja/docs/Learn/Getting_started_with_the_web/JavaScript_basics
- [12] PHP (プログラミング言語) | Wikipedia
[https://ja.wikipedia.org/wiki/PHP_\(%E3%83%97%E3%83%AD%E3%82%B0%E3%83%A9%E3%83%9F%E3%83%B3%E3%82%B0%E8%A8%80%E8%AA%9E\)](https://ja.wikipedia.org/wiki/PHP_(%E3%83%97%E3%83%AD%E3%82%B0%E3%83%A9%E3%83%9F%E3%83%B3%E3%82%B0%E8%A8%80%E8%AA%9E))
- [13] オリジン間リソース共有 (CORS) | MDN – Mozilla
<https://developer.mozilla.org/ja/docs/Web/HTTP/CORS>
- [14] Web Speech API | MDN – Mozilla
https://developer.mozilla.org/ja/docs/Web/API/Web_Speech_API
- [15] jQuery | Wikipedia
<https://ja.wikipedia.org/wiki/JQuery>
- [16] jQuery | jQuery
<https://jquery.com/>
- [17] 正規表現 – JavaScript | MDN – Mozilla
https://developer.mozilla.org/ja/docs/Web/JavaScript/Guide/Regular_Expressions
- [18] 青空朗読 | 青空朗読
<https://aozoraroudoku.jp/>