

Web プログラミング基礎

単語クイズを自動生成して
楽しみながら覚える
Web アプリケーションの考案

2023 年 1 月 30 日

K22120 牧 野 遥 斗

1 目的

このアプリケーションは、日々の授業や資格取得など数多くある単語を楽しんで覚えるために使う Web アプリケーションである。我々は外国語の勉強をする時に数多くの単語を覚える必要がある。英語だったり、フランス語だったり、日本語だったり、その他の言語でも同じだ。しかし、単語を覚えるのはとても大変である。単語を覚えるには、単語帳を使うのが一般的だ。しかし、単語帳を使うと、単語を覚えるのに時間がかかる。また、単語帳を使うと、単語を覚えるのに面白くない。そこで、我々は、単語を覚えるのに面白みを持たせることで、単語を覚えるのに時間を短縮することを目的としている。

2 機能

このアプリケーションは、単語クイズを自動生成して楽しみながら覚えることができる。そのため、単語を登録する編集画面と、単語をクイズするクイズ画面がある。編集画面では、単語とその意味を登録することができる。クイズ画面では、単語クイズを自動生成して楽しみながら覚えることができる。各画面の詳細は、図 1 と図 2 に示す。



図 1 編集画面



図 2 クイズ画面

2.1 編集画面

編集画面では、単語とその意味を登録することができる。単語とその意味を登録すると、クイズの自動生成ができるかどうかを判定する。

クイズの自動生成ができる単語は、今回 API で使用した Word2Vec の単語ベクトルに含まれている単語である。Word2Vec とは、文章中の単語を数値ベクトルに変換してその意味を把握する自然言語処理の手法である。数値ベクトルに変換を行っているため、演算処理をすることが可能で、 \cos 類似度を用いることで単語の近侍度を求めることができる。今回使用した Word2Vec のサンプルデータは日本語 Wikipedia のサイトを用いているため、日本語以外の単語は対応していない場合が多い。対応していない単語を用いると自動でクイズを生成することができないため、対応していない単語を登録されないようにエラー処理を行う。クイズの自動生成ができない単語のエラー画面は図 3 に示す。エラーを表示することにより、クイズの自動生成ができない単語を登録することを防ぎ、クイズの自動生成ができる単語のみを登録することができる。

覚えたい単語を登録したら、クイズ画面に移動することができる。左下の緑のボタン「クイズスタート」を押し、クイズ画面に移動する。



図3 クイズの自動生成ができない単語のエラー画面

2.2 クイズ画面

クイズ画面では、単語クイズを自動生成して楽しみながら覚えることができる。クイズは3択クイズで、単語の意味を選択することができる。クイズの自動生成は、Word2Vecの単語ベクトルを用いて行う。正解の単語の意味に近い単語を50件取得し、その単語の中から二種類選択肢として表示する。クイズの選択画面は図4に示す。

クイズは、回答をすると正誤に合わせた効果音を流すようにし、ユーザーエクスペリエンスを高めるように設計を行なった。最後のクイズが終ると、クイズが終了したことをアナウンスする。クイズの終了画面は図5に示す。



図4 クイズの選択画面

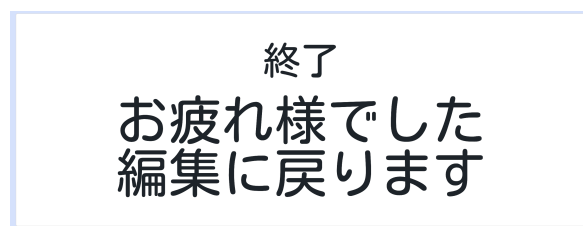


図5 クイズの終了画面

3 目的

このアプリは、単語を覚えるのに面白みを持たせることで、単語を覚える時間を短縮することを目的としている。そのため、単語を覚えるのに面白みを持たせるために、単語をクイズ形式にし

た。そして、クイズを作成する時に、自分ですべての問題を作るというのは大変である。そこで、単語の意味を入力するだけで、クイズを自動生成することができるようにした。クイズの自動生成には Word2Vec を用いることで、単語の意味を入力するだけで、単語の意味に近い単語を取得することができ、簡単にクイズを作成することができる。

機能面は、単語を登録し、クイズをするという中核をなす部分は完成している。エラー処理なども行うことで、ユーザーがエクスペリエンスを上げることに成功し、満足のいくアプリケーションを作成することができた。しかし、きめ細かい機能を追加することができなかったので、次の節では改善点を述べていく。

3.1 改善点

今後の改善点として、以下のようなものがある。様々な単語を登録することで、様々なジャンルの単語が乱立してしまい覚えたい単語以外の単語もクイズに出てきてしまう問題点がある。この問題点を解決するために、単語を登録する際に、単語のジャンルを登録することができるようにする。単語のジャンルを登録することで、単語のジャンルを指定してクイズを出題することができる。

次に、Word2Vec の単語ベクトルを用いてクイズを自動生成しているため、単語ベクトルにない言葉はクイズに出題できない。今回は Wikipedia の記事をサンプルとして使用しているため、日本語以外の言語は単語ベクトルにあまり含まれていない。この問題点を解決するために、英語版の Wikipedia をサンプルとして使用することで、単語ベクトルに含まれていない単語もクイズに出題することができる。また、中国語やフランス語にも対応させるため、世界各国のサンプルデータを取得することで、単語ベクトルに含まれていない単語もクイズに出題することができる。

4 付録

4.1 API 通信部分

今回 Word2Vec と通信を行うために、axios を用いて API 通信を行なった。axios を用いるメリットとして、使いやすく軽量である点、自動的な JSON データ変換を行う点、Promise API にサポートしている点などがあるため本プログラムに採用した。axios を使用しているコードは図 6 に示す。

```
axios.get(`https://word2vec.harutiro.net/near?get_number=50&str=${question.read}`)
  .then(response => {
    if ("OK" == response.data.status) {
      nearList.push(response.data.data[Math.floor(Math.random() * 50)]);
      nearList.push(response.data.data[Math.floor(Math.random() * 50)]);

      nearList.sort(() => Math.random() - 0.5);

      document.querySelector("#ans-1").innerText = nearList[0];
      document.querySelector("#ans-2").innerText = nearList[1];
      document.querySelector("#ans-3").innerText = nearList[2];
    } else {
      console.log(response.data)
    }
  })
  .catch(error => console.error(error))
```

図 6 axios を使用しているコード

4.2 データクラスについて

今後コードを更新するにあたり、わかりやすいソースコードを書く必要がある。ソースコードをわかりやすく書くためには、明示的にデータを定義する必要がある。そのため、データクラスを擬似的に定義を行い、ソースコードの可読性を高めた。図 7,8 は、データクラスの定義部分と使用部分を示している。

```
// データクラス
function Word(_id, _write, _read, _tag) {
  this.write = _write;
  this.read = _read;
  this.id = _id;
  this.tag = _tag;
}
```

図 7 データクラスの定義部分

```
// リストを新規追加
addItem(write, read, id) {
  this.DOM.containerRow.appendChild(this._createItem(write, read, id));
  wordList.push(new Word(id, write, read, "tag"));
  console.log(wordList);

  localStorage.setItem("wordList", JSON.stringify(wordList));
}
```

図 8 データクラスの使用部分

4.3 デプロイについて

今回のシステムは、Github の Pages を用いてデプロイを行なった。Github の Pages を用いるメリットとして、無料で簡単にデプロイを行うことができる点、Github のリポジトリにあるソースコードをそのままデプロイすることができる点などがあるため本プログラムに採用した。また、URL を独自ドメインに変更することで、より親しみやすい URL に変更することができる。

下記 URL からアクセスできる。<https://word-list-web.harutiro.net/>

4.4 css について

今回、わかりやすい UI を作成するため、クラウドベースのユーザーインターフェイスデザインツールである、Figma を用いて UI 設計を行なった。UI 設計をするにあたり、マテリアルデザインに準拠をしどんなユーザーにも直感的に操作ができるようなデザインを心がけた。

統一されたデザインを作成するため、bootstrap を用いて開発を行なった。class を使用するだ

けで簡単に css を記載することができるためとても便利である。図 10 は、bootstrap を用いて css を記載している例である。



図 9 Bootstrap の公式ページ

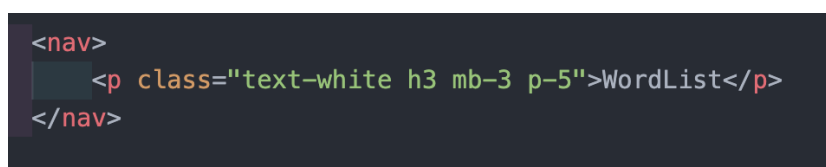


図 10 Bootstrap を用いて css を記載している例

4.5 favicon について

favicon は、ブラウザのタブに表示されるアイコンである。今回は、favicon を作成するために、Figma を使用した。Figma を使用するメリットとして、簡単にアイコンを作成することができる点、アイコンを作成する際に、デザインを統一することができる点などがあるため本プログラムに採用した。図 11 は、Figma を使用して作成した favicon である。



図 11 Figma を使用して作成した favicon