Web プログラミング及び演習

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

情報科学部 情報科学科 コンピュータシステム専攻
K22120 牧野遥斗

2025 年 1 月 22 日

目次

1. 目標

このアプリケーションは、日々の授業や資格取得に必要な単語を楽しく効率的に覚えるための Web アプリケーションである。言語学習において多くの単語を覚えることは不可欠だが、従来の単語帳を使った暗記法では、時間がかかるうえに退屈さを感じる。そこで、このアプリはゲーム性を持たせたクイズ形式を導入し、学習の楽しさを高め、単語暗記の効率を向上させる。

2. 目的

このアプリケーションの目的は、単語暗記の効率化と学習の楽しさの両立である。その実現のため、以下の工夫を取り入れた。

1. クイズ形式で単語を覚える仕組み: 単語暗記に楽しさを加え、学習へのモチベーションを高める
2. クイズの自動生成機能: Word2Vec を活用し、ユーザーが単語の意味を入力するだけで、それに関連する単語を基にクイズを自動生成する。この機能により、クイズ作成の手間を大幅に削減した。

さらに、単語登録やクイズプレイの基本機能に加え、エラー処理を充実させ、ユーザー体験 (UX) の向上にも取り組んだ。現時点では中核機能が完成しており、ユーザーが満足するアプリケーションを提供できているが、さらなる改善点については次節で述べます。

3. 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。

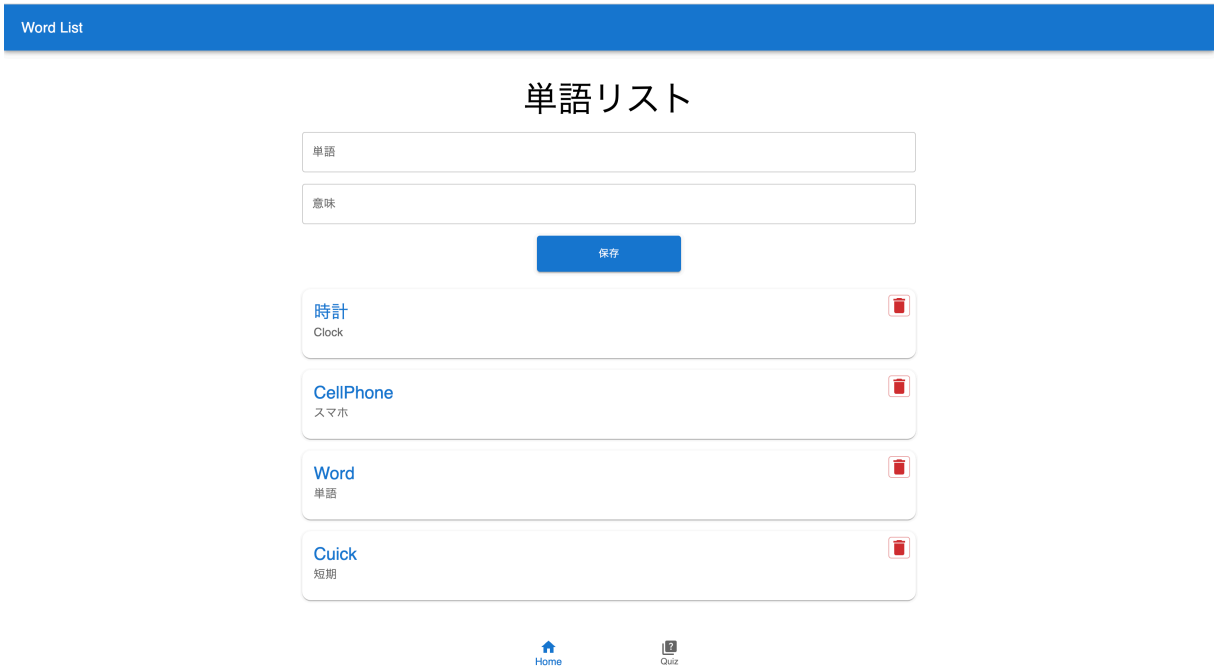

図 1: アプリ画面

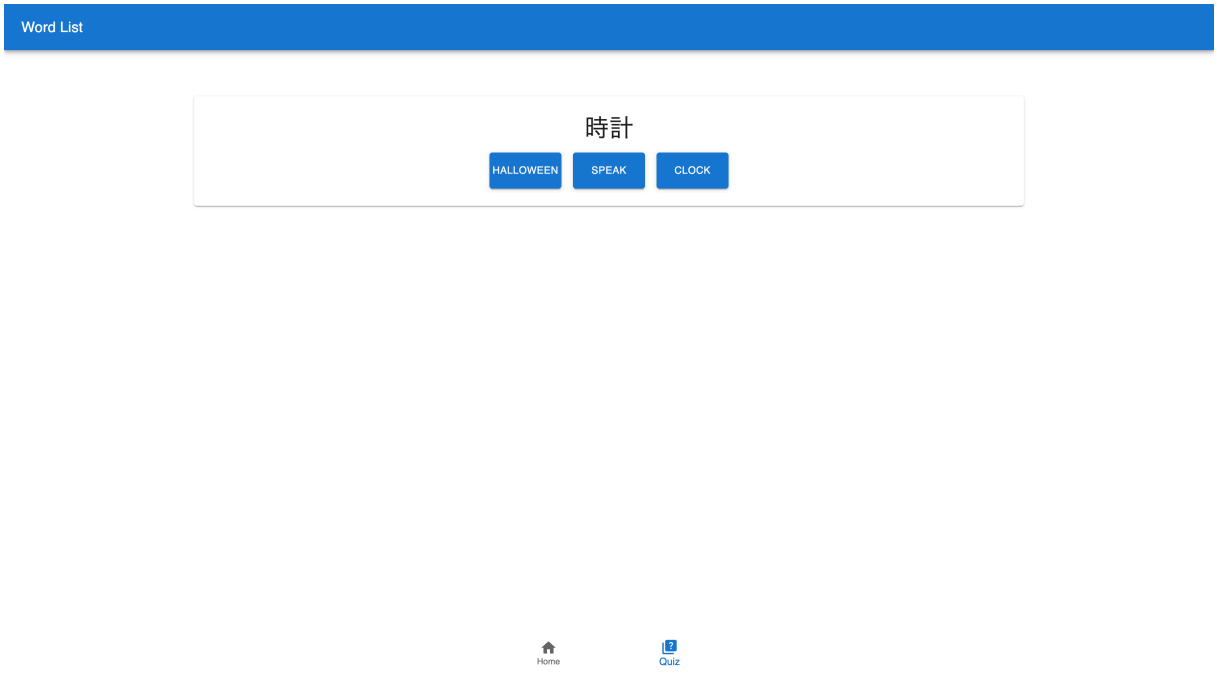

図 2: クイズ画面

3.1. 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。

図 3: 生成に使えない文字のエラー画面

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。ボトムナビゲーションにある「Quiz」を押し、クイズ画面に移動する。

3.2. 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 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: クイズの選択ボタン

図 5: クイズ終了時の画面

4. データ構造について

図 6 は、WordList テーブルと QuizResult テーブルの関係を示す。WordList テーブルと QuizResultテーブルは、これらのデータ構造がどのように設計され、相互に関連付けられているかを視覚的に表している。

WordList			
UUID	id	PK	単語の一意の識別子
VARCHAR	write		単語
VARCHAR	read		単語の意味

QuizResult		
INTEGER	correctAnswers	正解数
INTEGER	totalQuestions	問題数
TIMESTAMP	timestamp	クイズ結果の記録日時

図 6: ER 図

図 6 に示す通り、WordList テーブルと QuizResult テーブルは明確に分離されており、それぞれ異なるデータの役割を果たしている。この設計により、単語リストの管理とクイズの結果記録を効率的に行うことができる。以下のように、WordList テーブルと QuizResult テーブルのマイグレーションファイルとモデルファイルを作成し、データベースの設計を行った。

4.1. WordList テーブル

主キー (id) を持ち、単語リストの基本情報 (単語の表記とその意味) を管理する。 各レコードには、単語（write）とその意味（read）が格納される。 時間情報は timestamps (作成日時と更新日時) に含まれる。

4.2. QuizResult テーブル

正解数 (correctAnswers)、問題数 (totalQuestions)、およびクイズ結果の記録日時 (timestamp) を保持する。 クイズに関する統計データを保存するための役割を果たす。

4.3. リレーション

両テーブル間に直接的なリレーションは存在しないが、論理的な関係としてクイズに出題される単語リスト（WordList）とその結果（QuizResult）が独立した形で機能するよう設計されている。 このデータ構造の設計により、単語管理とクイズ結果の記録がスムーズかつ効率的に行えるデータベースを構築することができる。

5. 改善点・反省点

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。

6. 付録

6.1. 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 7 に示す。

```
useEffect(() => {
  // wordListが空のときにエラーメッセージを表示
  if (wordList.length === 0) return;

  if (question.read) {
    axios
      .get(`https://word2vec.harutiro.net/near?get_number=50&str=${question.read}`)
      .then((response) => {
        if (response.data.status === 'OK') {
          const nearList = [
            response.data.data[Math.floor(Math.random() * 50)],
            response.data.data[Math.floor(Math.random() * 50)],
            question.read,
          ];
          setQuestion((prev) => ({
            ...prev,
            answers: nearList.sort(() => Math.random() - 0.5),
          }));
        }
      })
      .catch((error) => console.error(error));
  }
}, [question.read, wordList]); // wordListが空でないかも確認
```

図 7: axios を使用して API を通信するコード

6.2. デプロイについて

今回のシステムは、CloudflarePages を用いてデプロイを行なった。CloudflarePages 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。

下記 URL からアクセスできる。 https://my-wordlist-app.pages.dev/

6.3. 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 8: favicon の作成