

文章要約・校正ツール

1. はじめに

本ツールは、Geminiを活用し、ユーザーが入力した文章を、ご要望やオプションに合わせて要約・校正して返すWebアプリケーションです。

プログラムがプロンプトを作成するため、ユーザーはただ文章を入力するだけで回答が得られます。これにより、利便性が向上するだけでなく、通常のGemini利用よりも適切な文章を手に入れることができます。

開発動機

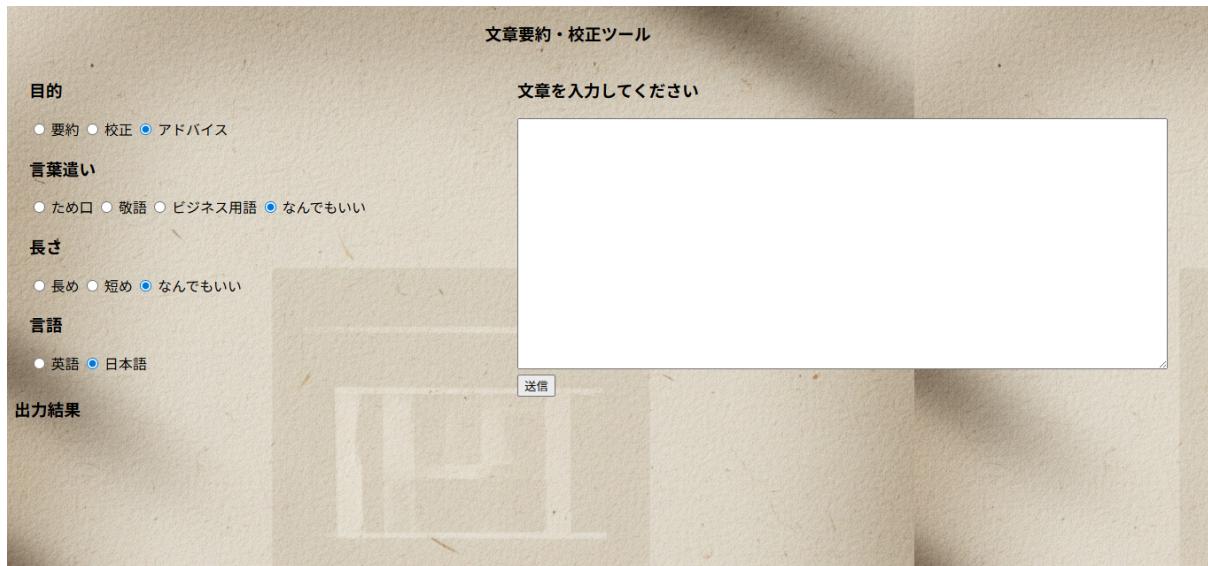
わたしは週報などを書く際、いつもGeminiに文章校正を依頼していましたが、Geminiが、わたしが予め教えたルールを守らなかったり、勝手に別のルールを追加したりすることがありました。そこで、自分でGeminiを利用するアプリケーションを作成すれば、細かなルール設定をわたしが行わなくともプログラムが全て処理してくれるため、より便利になるのではないかという発想に至りました。このアイディアを整理し、文章要約・校正ツールとして作成することにしました。

使用する言語・ライブラリ

- Python
- HTML+css
- JavaScript
- flask
- dotenv
- google-generativeai

2. 画面・使い方

Webアプリの画面構成はこちらになります。

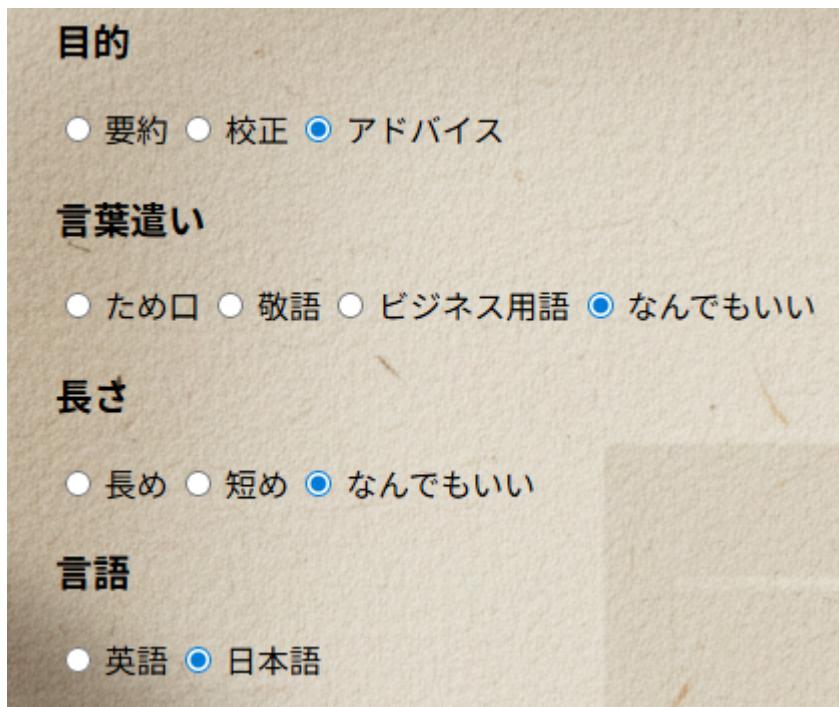


各部分について説明します。

【オプションパネル】

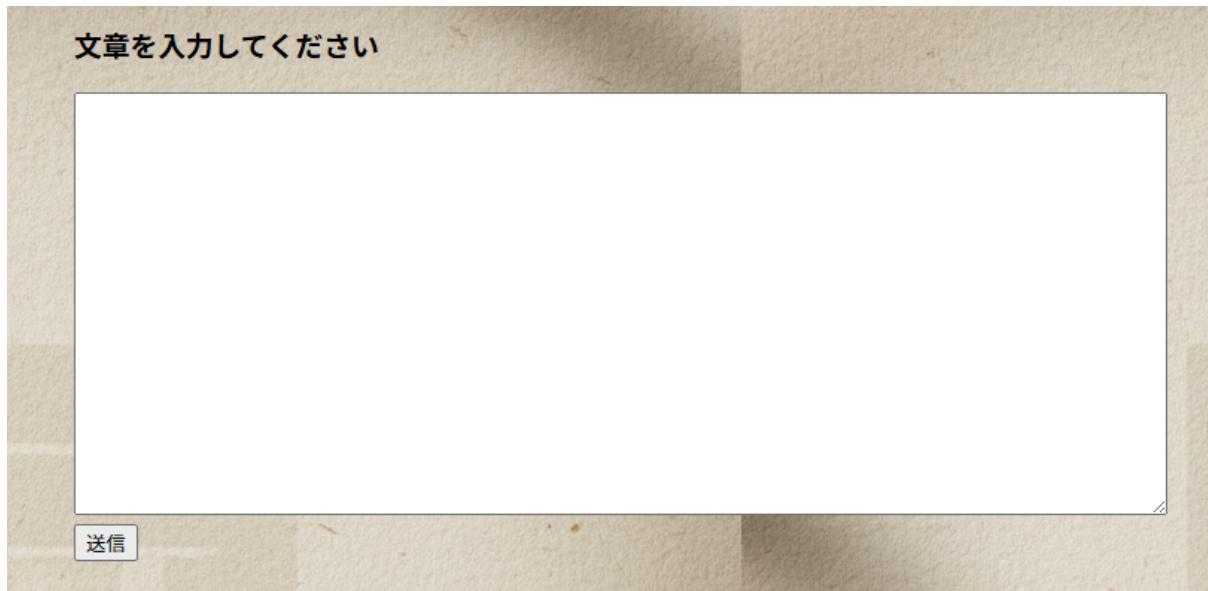
欲しい回答のオプションを選べます。詳しくは後程紹介します。

デフォルトは日本語でのアドバイスのみになっています。



【文章入力エリア】

文章を入力して送信を押したら、プログラムが文章と選んだオプションに基づいてプロンプトを作成しGeminiに送信して、回答を表示します。



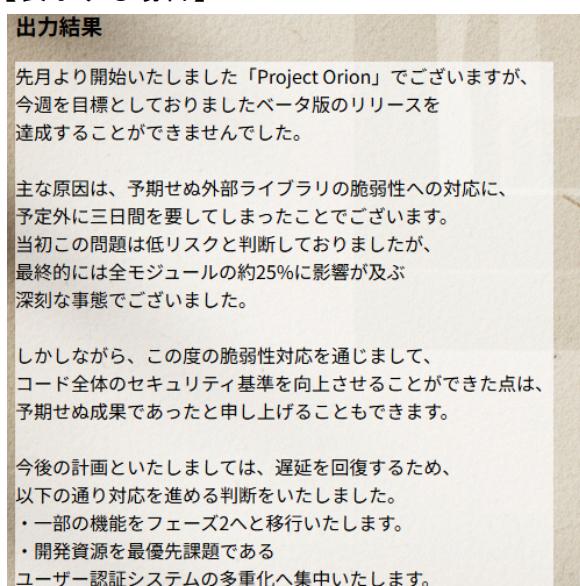
【出力エリア】

Geminiの回答を表示します。基本的には何も表示されませんが、回答を表示したら文章は半透明な白エリアに囲まれます。

【表示なしの場合】



【表示する場合】



3、オプション・プロンプト説明

オプションエリアの各オプションについての説明は以下です。

オプション	説明
目的	入力した文章に対するどんな回答を求めるか選べる、要約/校正/アドバイスで選択できる
言葉遣い	ユーザーが回答の言葉遣いを選べる、ため口/敬語/ビジネス用語/指定なしで選択できる
長さ	ユーザーが回答の長さを選べる、長め/短め/指定なしで選択できる
言語	回答の使用する言語が選べる、日本語/英語で選択できる

プログラムはユーザーが選んだオプションと入力した文章と以下の句でプロンプトを作成します。

オプション	説明
文章だけで返事してください。	「かしこまりました。」などのGeminiが回答の先頭に入れる文章が要らないため出力させない
装飾的なテキスト、導入文、結論、見出しを一切含めないでください。	回答に要らないため出力させない
元の文章が伝える意図とニュアンスを完全に維持してください。	Geminiが勝手に元の文章のニュアンスを変えさせないように
"や*などの記号を使わないでください。	太文字の**などを使わせない
箇条書きには・を使用してください。	箇条書きする場合ポイントを見やすくするため
適切にhtmlの改行を入れて改行してください。長くても50文字ぐらいで改行してください。	 で改行させて、innerHTMLで自動的に回答は改行し、文章は見やすくなる

実際のプロンプトはこんな感じになります。

以下の文章を、以下の条件で日本語で校正してください。

- ・言葉遣い:なんでもいい
- ・長さ:なんでもいい
- ・文章だけで返事してください。
- ・装飾的なテキスト、導入文、結論、見出しを一切含めないでください。
- ・元の文章が伝える意図とニュアンスを完全に維持してください。
- ・"や*などの記号を使わないでください。
- ・箇条書きには・を使用してください。
- ・適切にhtmlの改行を入れて改行してください。
- ・長くとも50文字ぐらいで改行してください。

最近、私たちの会社は、あたらしい企画をスタートする事になりました。その目的は、顧客のエンゲージメントを増やす事。このプロジェクトは、多大なリソースを投入する事が必須であると、私は考える所存です。特に、メンバーのモチベーションがあがりそうにないので、そこが一番の懸念であると言える。先ず、我々は来週まで、最終的な報告書を纏める必要があり、そしてその後で上層部にプレゼンするだろう。しかし、その計画のスケジュールは、ちょっと難題がたくさん有るのです。

校正した結果はこちらになります。

最近、私たちの会社では新しい企画が始まりました。
その目的は、顧客エンゲージメントを向上させることです。
このプロジェクトには、多大なリソースの投入が
必須だと考えています。
特に、メンバーのモチベーション維持が一番の懸念です。
まず、来週までに最終報告書をまとめる必要があり、
その後、上層部へプレゼンテーションを行う予定です。
しかし、計画のスケジュールには課題が多くあります。

4. システムの流れ

ユーザーが文章を入力して送信を押してから回答が表示されるまでのプログラムの流れを紹介します。

プロセス	説明
1、オプション・文章の取得	JavaScriptで各ラジオボタンとテキストエリアのvalueプロパティを取得する
2、プロンプトの作成	取得したvalueと固定の文字列を合わせて、Geminiに送信するプロンプトを作成
3、app.pyの関数を呼び出す	JavaScriptのfetch関数でapp.pyにある関数を呼び出し、Python側はリクエストからプロンプトを取得し、Geminiに接続しプロンプトを送信
4、結果をキャッチし表示する	Geminiの回答をjson文字列の戻り値で戻し、fetch関数はそれをキャッチし出力結果のdivのinnerHTMLに代入する

5. 苦労点・工夫した点

製作時に苦労した点と工夫した点について紹介します。

・HTMLとCSSでWebアプリを構築するにあたって、JavaScriptでPythonファイルの関数を呼び出してGeminiに接続しようとしたが、fetch関数では**Same-Origin Policy**でオリジンが異なるファイルを呼べないという問題があったため、Flaskを使用し、HTMLファイルとPythonファイルを同じローカルサーバーで動作させることで、無事呼び出せるようになりました。

・わたしは専門学校でHTML、CSS、JavaScriptを勉強したため、Flaskのみに頼らず、自身でWebアプリをデザインしてみました。ボタンやテキストエリアの配置を調整し、Geminiに背景に相応しい画像を生成してもらい、それを背景画像に使用しました。また、JavaScriptを使用することでラジオボタンによるオプション選択が可能となり、よりユーザーが求める回答を取得できるようになりました。

・プロンプトエンジニアリングは初めてだったため、最初は普段Geminiに文章校正を依頼する際のようなプロンプトで試しましたが、中途半端な回答になりました。そこで、Geminiにアドバイスを聞いた結果、様々なオプションを入れることを勧められました。試行錯誤を重ねた結果、現在のプロンプトに至り、満足できる回答を得ることができました。

現状できていないこと・将来追加したい機能

・無料のGemini APIを使用しているため、トークンの上限に到達することが多いです。文章とプロンプトを合わせるとかなり長くなるため、1日数回送信するだけで使用不可になってしまいます。資金があれば、いつか課金して自由に利用できるアプリケーションにしたいと考えています。

・制作の予定時間の都合上、オプションを四つだけ設定しました。今後時間ができたら、一人称や文章の読み手などのオプションを追加する予定です。