Implementing GPT-4 in the Portuguese Learning Application

Overview

This document outlines the integration of the GPT-4 API into the Portuguese learning application. The goal is to process user inputs, classify them, and generate structured responses that align with predefined types and interfaces that I previously created and showed you while making the chat interactive.

AI Response Structure

The AI responses must follow a specific structure to ensure consistency in the UI and backend logic. Below are the key types and interfaces:

Response Types

- `text` - Simple text response
- `question` - AI-generated questions (Multiple Choice or Fill in the Blanks)
- `correction` - AI provides a correction to user input
- `hint` - AI gives a hint for a question
- `explanation` - AI explains a concept
- `feedback` - General feedback on user performance

Question Types

- `MultipleChoice` - The user selects an answer from predefined options.
- `FillInTheBlanks` - The user fills in missing words within a sentence.

Message Senders

- `User` - Messages sent by the user.
- `AI` - Messages generated by the AI.

Implementation Plan

Step 1: Process User Input

- The frontend captures the user input and sends it to the backend.
- The backend forwards the input to the GPT-4 API, along with conversation history for context preservation.

Step 2: Classify the Input

- The backend analyzes the input using natural language processing.
- It classifies the input into one of the predefined `ResponseType` categories.

Step 3: Generate AI Response

- If the response type is `question`, the backend generates a structured question object (`MultipleChoice` or `FillInTheBlanks`).
- If the response type is `correction`, `hint`, or `explanation`, the AI provides feedback in a `TextResponse` format.
- The AI response follows the `AIChatResponse` structure.

Step 4: Send Response to Frontend

- The backend returns the structured response to the frontend.
- The frontend parses the response and displays the appropriate UI elements.

Step 5: Handle User Interaction with Questions

- If the response contains a question, the user interacts with it by selecting an option or filling in blanks.
- The answer is validated against the `correct_answers` field in the question object.

Step 6: Send User Answer for Evaluation

- The frontend sends the user's answer, along with the question ID and metadata, to the backend.
- The backend evaluates the answer and returns structured feedback.
- The feedback indicates correctness and may include additional explanations.

Step 7: Continue the Interactive Learning Cycle

- The interaction continues as the AI provides more questions, explanations, and feedback.
- This structured approach ensures a seamless learning experience.

Plan Summary

The backend will analyze the input using natural language processing and classify it into one of the predefined `ResponseType` categories (e.g., `text`, `question`, `correction`, etc.).

- If the response type is `question`, the backend will generate a question object based on the `QuestionTypes` (`MultipleChoice` or `FillInTheBlanks`) and structure it according to our `QuestionResponse` interface.
- If the response type is `correction`, `hint`, or `explanation`, the backend will provide feedback using a `TextResponse` formatted message.
- The response will be returned to the frontend, maintaining the structure of `AIChatResponse`, ensuring consistency in the UI.
- The frontend will parse the response and display the appropriate UI elements, such as multiple-choice options, fill-in-the-blank inputs, or simple text messages.
- If the response contains a question, the user will interact with it, and their answer will be validated against the `correct_answers` field in the question object.
- The user's answer, along with the question ID and additional metadata, will be sent back to the backend for evaluation.
- Based on the evaluation, the backend will return feedback in the form of another structured `AIChatResponse`, indicating whether the answer was correct and providing further explanation if needed.
- This cycle will continue, allowing for an interactive and structured learning experience.

This document serves as a blueprint for ensuring a structured and efficient AI-driven learning process.

AI-Level Implementation Steps

To integrate ChatGPT-4 API effectively within our Portuguese learning application while maintaining structured responses, follow these steps:

---

1. Contextual Understanding & Memory Handling

- Enable **thread-based chat history** if using OpenAI's API with function calling to retain past interactions.
- Store user interactions on our backend to maintain **context** and **progress tracking**.
- Implement a **conversation state manager** to pass relevant past messages in the API request.

---

2. Input Processing & Classification

- When a user sends input, the backend must:
    - Use **Natural Language Processing (NLP)** to classify it into one of the predefined `ResponseType` categories:
        - **Text** (general response)
        - **Question** (learning interaction)
        - **Correction** (fix user mistakes)
        - **Hint** (help user without giving away the answer)
        - **Explanation** (detailed concept clarification)
        - **Feedback** (performance evaluation)
    - Use **ChatGPT function calling** (or a classifier model) to analyze user input and determine the appropriate response type.

---

3. Generating AI Responses

- Depending on the detected `ResponseType`, the AI must generate the response in a structured format:
    - **Question Type** → Create a valid `QuestionResponse` object (either `MultipleChoiceQuestion` or `FillInTheBlankQuestion`).
    - **Correction, Hint, Explanation** → Return structured `TextResponse` with detailed feedback.
    - **Feedback** → Provide performance analysis based on user's historical responses.

Example function call for structured response:

```
{
  "function": "generate_chat_response",
  "parameters": {
    "responseType": "question",
    "questionType": "MultipleChoice",
    "difficulty": "medium",
    "questionText": "Which word means 'Hello' in Portuguese?",
    "options": ["Olá", "Adeus", "Obrigado", "Sim"],
    "correct_answers": ["Olá"]
  }
}
```

-

---

4. Structured AIChatResponse Formatting

The AI response should match the `AIChatResponse` interface:

```
{
  "id": 12345,
  "sender": "AI",
  "type": "question",
  "content": "Select the correct translation for 'Hello' in
Portuguese.",
  "payload": {
    "questions": [
      {
        "id": "q1",
        "type": "MultipleChoice",
        "questionText": "Which word means 'Hello' in Portuguese?",
        "questionDescription": "Choose the correct answer.",
        "correct_answers": ["Olá"],
        "difficulty": "easy",
        "options": ["Olá", "Adeus", "Obrigado", "Sim"]
      }
    ]
  }
}
```

-

---

5. Validating User Answers

- When the user responds, send the answer back to the AI/backend.
- Validate the response against the `correct_answers` field.
- If correct, return an **AIChatResponse** with positive reinforcement.
- If incorrect, return a **correction or hint**.

---

6. Continuous Learning & Adaptation

- Use **reinforcement learning** techniques to adjust response difficulty based on user performance.

- Track user **accuracy and mistake patterns** to personalize future AI-generated responses.

Backend Time Estimates (1 day = 8 working hours)

The frontend part is already there and when the backend is complete we'll just send messages back and forth

1. **Set up GPT-4 API integration** – 1 day
2. **Implement input classification using GPT** – 3 days
3. **Implement structured response generation** – 5 days
4. **Implement user answer validation via GPT** – 4 days
5. **API route development & request handling (Postman testable)** – 2 days
6. **Optimize, test, and debug responses** – 5 days

Total Estimate: 20 Days (Approximately 3 Weeks)