

Web Intelligence 第13回課題レポート

2025年7月11日
学籍番号：35714121
名前：福富隆大

使い方

Pythonファイルの実行方法

1. 基本的な実行方法

```
python ファイル名.py
```

2. 仮想環境を使用する場合

```
# 仮想環境の作成
python -m venv venv

# 仮想環境の有効化
# Windows:
venv\Scripts\activate
# macOS/Linux:
source venv/bin/activate

# 依存関係のインストール
pip install -r requirements.txt

# プログラムの実行
python math_detective_ai.py

# 仮想環境の無効化
deactivate
```

3. 環境変数の設定

```
# .envファイルを作成してAPI KEYを設定
echo "GEMINI_API_KEY=your_api_key_here" > .env
```

4. 必要なライブラリのインストール

```
pip install google-generativeai python-dotenv pillow
```

5. 実行例

```
# 現在のディレクトリで実行
python math_detective_ai.py

# 対話開始後の使用例
> x^2 + 3x + 2 = 0を解いてください
> 画像: math_problem.png
> ヒント
> 答え: x = -1, -2
```

必須課題：答えを直接言わず、ヒントだけを提供する学習支援AI

自己評価：A

プログラムの説明

Gemini APIを活用した対話型の数学学習支援システム。探偵キャラクターが3段階のヒント（概念的→手法的→具体的）を提供し、学習者の理解を段階的にサポートする。

工夫点

- 探偵キャラクターによる親しみやすい対話
- 厳格なシステムプロンプトで答えの直接提示を防止
- 段階的ヒントレベルによる適応的学習支援
- 会話履歴の管理による文脈理解

考察

LLMの自然言語処理能力を活用することで、従来の固定的なヒントシステムより柔軟で教育的な支援が可能となった。システムプロンプトによる制約が効果的に機能し、答えを直接提示せずに学習を促進できている。

発展課題(A)：Web検索を活用したヒント生成

何日か試行錯誤したのですが、自分ではAPIエラーを解決できなかったため、実装できませんでした。googleの検索結果を利用しなくても、十分に良いヒントを提供できるようになっていると思います。

発展課題(B)：学習効果を上げる手法の提案

自己評価：A

プログラムの説明

本プログラムは、Geminiモデルを活用し、生徒の不正解に対して**多角的かつ個別化されたフィードバック**を提供します。これにより、誤りの原因を具体的に分析し、生徒の思考プロセス改善を促すことを目指します。

```
prompt = f"""
{self.system_prompt}

問題: {self.conversation_history.problem}
これまでのヒント: {chr(10).join(self.conversation_history.hints_given)}
学生の答え: {user_answer}
```

学生の答えが正しいか判断し、以下のルールに従ってフィードバックを提供してください。
答えは絶対に直接教えず、学生の思考プロセスを改善するための示唆を与えてください。

【正解の場合】

- 🎉 で始まる祝福メッセージ
- 解法の要点を簡潔に説明
- 使用したヒント数: {len(self.conversation_history.hints_given)}
- (任意) 関連する発展的な概念や、別の解法があれば簡潔に示唆。

【不正解の場合】

- 😞 で始まる励ましメッセージ
- **間違っている原因を具体的に分析し、以下のいずれかまたは複数の観点からフィードバックを提供してください。**
 - **概念理解の不足**：どの数学的概念（例：二次関数の頂点の意味、ベクトルの内積の定義）について再確認が必要か。
 - **計算ミスの可能性**：特定の演算（例：符号、分数、括弧の展開）で注意すべき点。
 - **解法アプローチの誤り**：問題の種類に対して選んだ解法が適切でない、または手順が間違っている可能性。
 - **知識の関連付け**：関連する別の公式や定理を思い出そうに促す。
 - **具体的な修正点への示唆**：答えを直接言わずに、どこを再計算・再検討すべきか具体的な示唆を与える。
- 再考を促す励まし。
- (任意) もし該当する概念や解法について**補足情報（例えば、Web検索で見つかるような追加の解説記事や類似問題）があれば、それを見つけるためのキーワードや学習の方向性を簡潔に提案**してください。

フィードバックの例：

「😞 計算ミスをしている可能性があります。特に〇〇の部分を見直してみてください。△△の概念を再確認すると良いかもしれません。」

または

「😞 解法のアプローチが少しずれているようです。このタイプの問題では、まずは〇〇の公式を適用することを検討すると良いでしょう。」

.....

工夫点

- **概念理解の不足、計算ミス、解法アプローチの誤り**など、誤答の具体的な原因を分析するようプロンプトを工夫しました。これにより、生徒の弱点に合わせた**個別最適な学習示唆**が可能になります。

考察

この手法は、単なる正誤判定を超えて生徒の**深い理解**と**自律的学習**を促進し、全体的な学習効果を飛躍的に向上させると考えられます。

発展課題(C)：AIアプリケーションの評価指標

自己評価：A

プログラムの説明

本プログラムでは、数学学習支援AIの効果を定量的に測定するため、**教育効果指標**と**パフォーマンス指標**の2つの観点から包括的な評価システムを実装しました。

評価指標の設計

1. 教育効果指標

- **ヒント利用効率**: $\text{ヒント数} / \text{問題難易度}$ で学習者の理解速度を測定
- **解答時間**: 問題提示から正解までの時間で学習効率を評価
- **自立学習促進度**: 段階的ヒント提供により最小限のサポートで解決できたかを評価

2. パフォーマンス指標

- **正解率**: 提出された答えの正確性
- **セッション継続率**: 学習者の学習意欲・満足度の間接指標
- **エラー処理率**: システムの安定性・信頼性指標

実装された評価機能

```
# ConversationHistoryクラスで学習データを記録
@dataclass
class ConversationHistory:
    problem: str
    user_responses: List[str]
    hints_given: List[str]
    hint_level: int
    understanding_score: float
    start_time: float = 0.0
    end_time: float = 0.0

# check_answer メソッドで評価レポート生成
end_time = time.time()
time_taken = end_time - self.conversation_history.start_time
hints_used = len(self.conversation_history.hints_given)

is_correct = "🎉" in evaluation # 正解の場合のキーワードで判定

feedback_message = ""
if is_correct:
```

```
feedback_message = "🌟 **評価レポート** 🌟 \n" \
    f"✅ **正解！** おめでとうございます！ \n" \
    f"💡 使用ヒント数: {hints_used}回 \n" \
    f"🕒 解答時間: {time_taken:.2f}秒"
```

工夫点

- **リアルタイム評価:** 各問題解決後に即座に学習効果指標を表示し、学習者の自己認識を促進
- **多面的評価:** 単なる正誤判定ではなく、学習プロセス全体を評価対象とすることで、真の学習効果を測定
- **データ駆動型改善:** 蓄積された評価データを基に、ヒント提供戦略の最適化が可能
- **学習者フィードバック統合:** 評価指標を学習者に可視化することで、メタ認知能力の向上を支援

考察

このような包括的評価指標により、AIアプリケーションの教育効果を客観的に測定できます。特に**ヒント利用効率**と**解答時間**の組み合わせにより、学習者の理解度と学習スタイルの両方を把握でき、個別最適化された学習支援の実現に寄与します。また、これらの指標は教育現場での導入判断や改善方針決定において重要な根拠となります。

今後の発展可能性

現在の実装では問題ごとの評価指標計算にとどまっていますが、**データ永続化機能**（JSON/データベースへの保存）を追加することで、以下のような高度な評価・分析が可能になります：

- **長期学習進捗追跡:** 学習者の成長過程を時系列で分析
- **複数問題間での比較分析:** 分野別・難易度別の得意不得意パターンの特定
- **統計的学習効果測定:** 大量データに基づくより信頼性の高い評価指標
- **個別学習推奨システム:** 過去の学習データに基づく最適な問題・ヒント提示戦略

このような機能拡張により、単なる学習支援ツールから**適応的学習プラットフォーム**への発展が期待でき、より効果的な教育支援が実現可能となります。

発展課題(D)：テキスト以外の入力にも対応した機能

自己評価：A

プログラムの説明

Gemini 1.5 Flashの**マルチモーダル機能**を活用し、画像から数学問題を自動抽出して学習支援を行う機能を実装しました。学習者は手書きの問題や教科書の写真を撮影するだけで、AIが問題を認識し、従来のテキスト入力と同様の学習支援を受けることができます。

実装した機能

1. 画像入力インターフェース

```
# ユーザー入力処理の拡張
elif user_input.lower().startswith('画像:') or
user_input.lower().startswith('image:');
```

```
image_path = user_input.split(':', 1)[1].strip()
return self.process_image_problem(image_path)
```

2. 画像解析・問題抽出システム

```
def process_image_problem(self, image_path: str) -> str:
    # PIL Imageで画像を読み込み
    image = Image.open(image_path)

    # Geminiのマルチモーダル機能で問題抽出
    response = self.model.generate_content([prompt, image])
    extracted_problem = response.text.strip()

    # 抽出した問題を既存フローに統合
    return self.process_problem(extracted_problem)
```

3. 高精度OCR機能

- 手書き文字・印刷文字の両方に対応
- 数式記号・分数・ルート記号等の正確な認識
- 不鮮明な画像に対する推測補完機能

4. エラーハンドリング・ユーザビリティ

- ファイル存在確認・形式チェック
- 問題抽出失敗時の分かりやすいガイダンス
- サポート形式：JPG, PNG, GIF, BMP, WebP

工夫点

- **Geminiネイティブ統合**: 外部OCRライブラリを使わず、Geminiの画像認識能力を直接活用することで、数学的文脈を理解した高精度な問題抽出を実現
- **シームレスな統合**: 画像から抽出した問題を`process_problem()`メソッドに渡すことで、既存のヒント提供・評価システムをそのまま利用
- **包括的エラーハンドリング**: ファイル不存在、画像破損、問題抽出失敗など様々なエラーケースに対応した親切的なユーザーガイダンス
- **直感的なUI**: **画像：[パス]**という分かりやすいコマンド形式で、技術的知識がなくても利用可能

考察

この実装により、学習者の利便性が飛躍的に向上しました。特に**スマートフォンで撮影した問題写真を直接利用**することで、紙の問題集や黒板の問題にも対応でき、実用性が大幅に向上しています。

Geminiのマルチモーダル機能の活用により、単純なOCRを超えて**数学的文脈を理解した問題抽出**が可能となり、従来のテキスト入力と同等の学習支援品質を維持できています。

将来的には音声入力、PDF文書対応、リアルタイム画像認識等の機能拡張により、より包括的な学習支援プラットフォームへの発展が期待できます。