

Chatgpt を利用した野球選手のための練習方法提案システムの試作

太田航平¹ 新谷虎松^{2*}

概要: 本研究では、大規模言語モデル ChatGPT-4 を活用し、野球選手の個人特性に応じて練習方法を提案するシステムを試作した。従来、練習指導はコーチの経験に依存しており、特に初心者にとっては効果的な練習法を主体的に得ることが困難であった。本システムでは、身長・体重・利き腕・守備位置・長所・課題 の 6 項目を入力とし、ChatGPT-4 が具体的な数値を含む練習メニューと関連キーワードを生成する。評価実験として、野球未経験者 5 名と経験者 5 名 にシステムを利用させ、UI、応答時間、応答の質を 5 段階で評価した。その結果、UI と応答時間は両群で高評価を得たが、応答の質は未経験者には有用性が認められる一方、経験者には不十分であり、より専門的かつ実践的な内容が求められることが明らかになった。本研究は、生成 AI をスポーツ分野の学習支援に応用した事例として意義を持ち、初心者には実用的かつ理解しやすい練習支援を提供できる可能性を示した。一方で、経験者向けの有効性を高めるためには、プロンプト設計の改良や専門知識の導入 が今後の課題である。

キーワード: ChatGPT, 野球練習支援システム, Python, スポーツ教育支援

A Prototype System for Training Method Recommendations for Baseball Players Using ChatGPT

Ota Kohei¹ Toramatsu Shintani^{2*}

Abstract: This paper presents a prototype system that applies ChatGPT-4 to personalized baseball training support. The system generates training menus by analyzing six user inputs: height, weight, dominant hand, fielding position, strengths, and weaknesses. Unlike conventional coach-dependent guidance, it provides concrete and easy-to-follow suggestions such as "100 swings," along with relevant keywords to enhance usability. An evaluation with five inexperienced and five experienced players assessed UI, response time, and training quality. Both groups rated UI and response time highly, while inexperienced players found the training suggestions useful. In contrast, experienced players reported insufficient specialization and practicality. These results demonstrate the potential of generative AI for sports training support, particularly in assisting beginners, while also highlighting the need for improvements in prompt design and domain-specific knowledge integration to meet the demands of advanced users.

Keywords: ChatGPT, Baseball Practice Support System, Python, Sports Education Assistance

1.はじめに

ChatGPT は、米国の OpenAI 社が開発した対話型の文章生成 AI であり、GPT (Generative Pre-trained Transformer) と呼ばれるニューラルネットワークに基づく大規模言語モデルを基盤としている。膨大なパラメータを有することで、人間が執筆したかのように自然な文章を生成でき、翻訳・要約・質問応答など多様な自然言語処理タスクに対応可能である。近年特に注目を集めている理由は、ユーザが任意のテーマを入力すると、短時間で文脈の一貫したまとまった文章を自動生成できる点にある。その対話体験は極めて自然であり、利用者はまるで人間を相手にしているかのような感覚を得ることができる。

GPT は、単語の出現確率に基づき次の語を予測する仕組みにより自然言語をモデル化しており、Transformer アーキ

テクチャを用いて事前学習 (Pre-training) と微調整 (Fine-tuning) を組み合わせることで高度な表現能力を獲得している。GPT 系列の研究は比較的長い歴史を有し、多くの企業や研究機関で応用されてきたが、一般社会における認知度は限定的であった。しかし、2022 年に公開された ChatGPT によって状況は一変し、生成 AI 技術が爆発的に普及・認知される契機となった。ChatGPT の初期版は 2020 年に公開された GPT-3 を改良した GPT-3.5 を基盤としており、およそ数千億規模のパラメータを学習している。この膨大なパラメータ数が高精度な自然言語生成を可能にしている点は、学術的にも技術的にも重要である。

開発元の OpenAI は 2015 年 12 月に設立され、イーロン・マスク氏を含む複数の著名投資家が設立に関与している。同社は特定領域に特化した「弱い AI」ではなく、汎用的な知的活動を可能とする AGI (Artificial General Intelligence) の実現を目標に掲げている。これまでに画像生成 AI 「DALL-E 2」や音声認識システム「Whisper」なども開発しており、生成 AI 技術の発展に大きく寄与してきた。

¹ 岐阜聖徳学園大学 (当時)

² 岐阜聖徳学園大学

Gifu Shotoku Gakuen University

*責任著者(Corresponding author): tora@gifu.shotoku.ac.jp

ChatGPT の応用範囲は極めて広く、質問応答、プログラミング支援、翻訳、要約、電子メールやレポートの作成、小説執筆、表計算処理など多岐にわたる。さらに、ユーザーが「小学生にもわかるように説明せよ」といった条件を与えることで、回答の抽象度を調整することも可能である。また、ユーザーが特定の役割（例：教師、コーチ、翻訳者など）を ChatGPT に割り当てることで、その役割に即した応答を行わせることができる。このようなプロンプト設計（指示文設計）は、ChatGPT の実用性を大きく左右する重要な要素である。

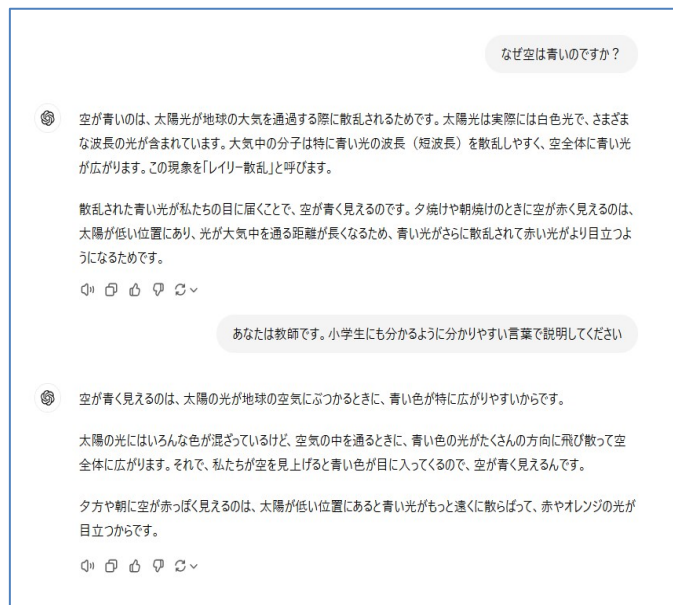


図 1.1 : ChatGPT に質問した例
使用モデル ChatGPT4o (11 月 7 日確認)

本研究の目的は、大規模言語モデル ChatGPT を活用して野球選手向けの練習支援システムを構築し、競技者のトレーニングを効果的に補助することである。従来、個人に最適化された練習方法を得るには指導者の経験や知識に依存する部分が大きく、選手自身が容易にアクセスできる仕組みは十分に整備されていなかった。本研究では、ChatGPT を Python 環境に導入し、アプリケーション上から対話的に利用できるようにする。

提案システムでは、利用者が身長・体重・利き腕・長所・課題といった個人特性を入力すると、その情報に基づき ChatGPT が練習方法を生成・提示する。これにより、従来のようにインターネット検索や一般的な情報に頼ることなく、より簡便かつ適切な練習提案を得ることが可能となる。さらに、生成される練習提案の有効性を評価実験によって検証し、本システムの必要性和実用的価値を明らかにすることを旨とする。

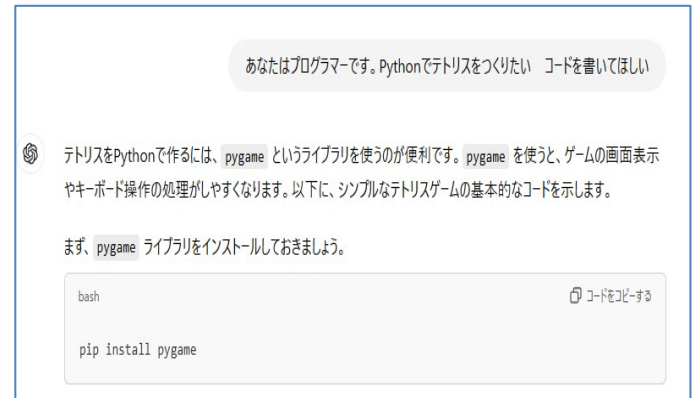


図 1.2 : プログラミングのサポートの例
使用モデル ChatGPT4o (11 月 7 日確認)

次に、テトリスゲームの基本的なコードを書きます。

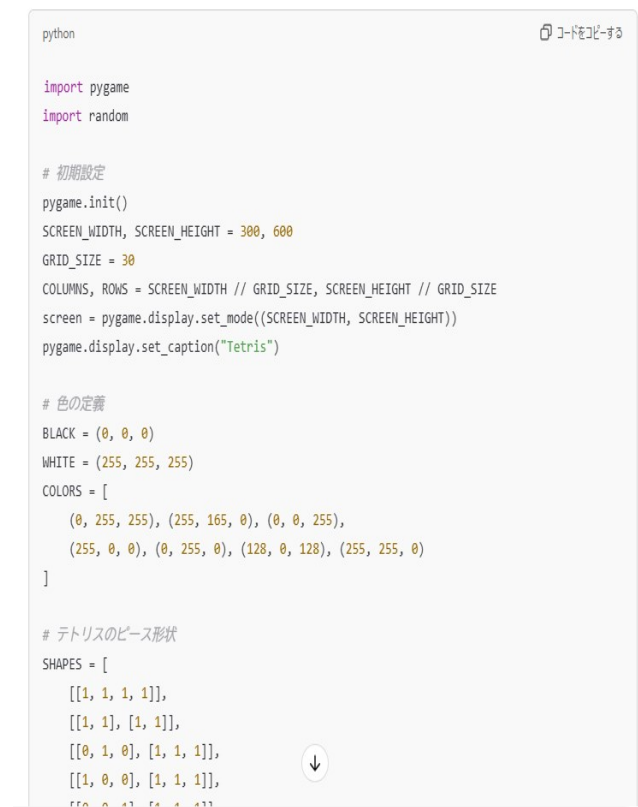


図 1.3 : 実際に表示されたコードの例
使用モデル ChatGPT4o (11 月 7 日確認)

本論文の以降の構成を以下に示す。2 章では、関連研究と本研究の位置付けについて述べる。3 章では、本研究の内容を示し、コードを記載していく。4 章では、評価実験とその結果について述べ、5 章では、評価実験の結果に基づいて本システムの考察と今後の展望について述べる。最後に 6 章で本論文をまとめる。

2. 関連研究

運用報告書における市況コメントおよび将来見通しは、投資家が資産の運用状況や今後の投資環境を理解するうえで重要な情報である。論文 [1] では、この課題に対して ChatGPT を活用し、市況コメントと見通しを自動生成するツールを開発している。手法としては、大規模言語モデル (LLM) の一つである ChatGPT-4 を用い、①資産リターンの算出、②変動要因記事の抽出、③変動要因の要約、④市況コメントの生成、⑤見通しコメントの生成、という一連のプロセスを通じて市況コメントと見通しを即時に生成する仕組みを構築している。

本研究で同論文を参考文献として取り上げた理由は、ChatGPT を活用した応用研究である点に加え、本研究で開発したシステムと同様に自動生成ツールの設計における有益な示唆を与えると考えられるためである。定性的評価の結果、本ツールで生成されたコメントは国内債券市場を除き概ね妥当性を有しており、記事情報を入力情報とすることで Hallucination (幻覚的出力) の抑制が可能であることが確認された。一方で、一部には事実と異なる記述も散見され、実用上は依然として人間によるファクトチェックの必要性が残ることも明らかとなった。

さらに定量的評価の結果からは、本ツールが人手で作成された市況コメントと類似度の高いコメントを自動生成できることが確認された。しかしながら、重要な記事の抽出方法には改善の余地があり、大量のテキストデータから適切な情報を抽出する仕組みを組み合わせることが今後の課題である。

近年、SNS の利用者は急速に増加しており、それに伴ってコミュニケーションの機会も拡大している。SNS は他者と容易に交流できる利点を持つ一方で、利用者を傷つけたり不快にさせたりする書き込みも増加している。その代表例が 誹謗中傷 であり、多くの場合、客観的に見ても不適切であることが明らかである。

論文 [4] では、この課題に対処するため、ChatGPT を活用した対話支援システムを提案している。本システムは、(1) 客観的な基準に基づく評価と、(2) 対話履歴から抽出した相手が好感あるいは不快感を示す文章特徴、という 2 種類の情報を用いて、ユーザーが送信しようとするメッセージに対し、必要に応じて対話相手に適応した修正案を提示するのである。

本研究でこの論文を参考文献に取り上げた理由は、ChatGPT を活用した研究である点に加え、プロンプト設計

に重点を置いている点にある。すなわち、ChatGPT にどのように指示を与えるかという点がシステム性能に大きく影響することを示しており、本研究におけるシステム設計にも有益な示唆を与えると考えられる。

システムの検証実験では、修正の要否判定については一定の精度が確認された。しかし、条件基準の設定や特徴量の選定には改善の余地が残されている。また、提示される修正案については、ユーザの視点ではなく対話相手側のメッセージとして生成されてしまうケースが多く、この点も今後の課題であると報告されている。

近年、旅行を計画する際にインターネットを利用して情報収集を行う機会が増加している。しかし、インターネット上には膨大な旅行情報が存在するため、具体的な目的地を決めずに情報を取捨選択することは容易ではない。論文 [3] では、この課題に対処するため、対話を通じてユーザの嗜好に合致する観光地を推薦するシステムを提案している。具体的には、ユーザが希望する観光地のイメージを入力すると、システムはその入力を形態素解析により必要な情報に変換し、分散表現としてモデルに入力する。また、行きたくない観光地のイメージも同様に分散表現化し、負の嗜好情報として扱う。これらの嗜好情報と負の嗜好情報の分散表現を加減算することでユーザの総合的嗜好を表現し、そのベクトルに最も近い観光地候補をコサイン類似度に基づいて提示する仕組みである。本研究において本論文を参考文献として取り上げた理由は、推薦システムの設計手法に関する知見を与えると考えられるためである。比較実験の結果、提案手法が従来手法に対して明確な優位性を示すことはできなかった。しかし、推薦候補地が限定的である場合や、分散表現の加減算が必ずしも期待通りに機能しない場合があるといった課題が明らかにされており、今後のシステム改善に有益な示唆を与えている。

データ分析における基本的手法の一つは比較であり、テキスト分析の分野においても複数のテキストデータを比較対象とする研究は数多く行われてきた。従来の方法では、テキスト中から特徴的な語を抽出し、その単語を基準として比較を行う手法が一般的である。しかし、この手法では抽出される単語の観点が一致しない場合があり、観点の異なる特徴同士を比較に用いることは困難である。論文 [7] では、この課題に対処するために ChatGPT を用いて比較の観点を生成し、それに基づいて二つの事柄の差差点および共通点を抽出するシステムを提案している。具体的には、入力として比較対象となる二つの事柄を与え、ChatGPT の API を通じて比較観点を自動生成し、その観点に従って差異と共通性を整理する仕組みを構築している。このために適切なプロンプト設計やアルゴリズムを構築することで、観点を揃えた比較を可能にすることを目指す

している。本研究でこの論文を参考にした理由は、ChatGPT を活用した支援システムの設計手法が示されており、筆者の研究である練習提案システムにおいても有用な知見を提供すると考えられるためである。実験の結果、本システムは比較を行う上で一定の妥当性を持つ出力を生成することが確認された。一方で、誤りを含む記述や解釈の不一致を招く出力、さらには同内容の繰り返しといった問題点も報告されており、これらの改善が今後の課題として指摘されている。

論文 [8] では、Jupyter Notebook を用いた対話的プログラミング演習支援に関する研究を行っている。Jupyter Notebook は学生が自主的に事前学習を行う環境を提供できる利点を有するが、一方で個別学習では疑問点を即座に質問できないという課題がある。教員やティーチングアシスタントへの質問対応は可能であるものの、指導者の経験や学生の理解度に依存して指導スタイルが大きく異なり、どの教授法が最適であるかを客観的に評価することは困難である。

この課題に対し、本論文では大規模言語モデル (LLM) を用いた教授方法の評価手法を提案している。具体的には、教授方法の単純化したモデルを構築し、ChatGPT をカスタマイズして複数の GPTs を作成することで、異なる教授スタイルを疑似的に実現した。提案手法では、プログラミング演習における学生との対話を再現し、以下の 3 種類の教授スタイルを設計・評価している。

- Type A : 解答例を即時に提示するスタイル
- Type B : 具体的なアドバイスを与えるが、初期段階では解答例を提示しないスタイル
- Type C : 考え方やアプローチのみを説明し、解答例を一切提示しないスタイル

これらの GPT アシスタントを用いて対話内容を分析することで、教授方法の特性を評価した。その結果、Type A は短期間で成果を求める状況に適し、Type B は思考力や実践的な学習の促進に有効であり、Type C は応用力や深い理解を養うには効果的であるが、習熟度の低い学習者には難易度が高いという知見が得られた。

本研究において本論文を参考とした理由は、ChatGPT を活用して教授スタイルをモデル化し、さらに GPTs を構築するアプローチを取っている点である。これは筆者が試作する練習提案システムにおいても参考になると考えられる。ただし、本研究の評価は主観の評価に留まっており、今後は実際の学生を対象とした対照実験を行い、教授方法の違いによる効果を定量的に検証する必要性が指摘されている。

近年、急速な発展を遂げている生成系 AI は教育現場にも大きな影響を及ぼしている。学習者は OpenAI が提供する ChatGPT や Google が提供する Gemini AI などを利用してプログラムの作成や課題に取り組む機会が増加している。しかし一方で、生成系 AI は質問に対して直接的な解答を提示することが多く、学習者が自ら疑問を考え、解決に至る機会を奪う可能性が指摘されている。

この課題を克服する一つの方法として、学習者の疑問に対して即座に答えを提示するのではなく、答えに至るためのヒントを提供することによって、学習者が主体的に思考し解決に至るプロセスを支援するアプローチが考えられる。論文 [4] では、この観点から、学習者の疑問を入力として受け取り、その解答を直接提示するのではなく、導出のためのヒントを返す対話型学習支援システムを提案している。本研究で同論文を参考にした理由は、対話型学習支援という特徴が、本研究で試作するシステムの設計にも応用可能であると考えられるためである。

論文 [4] のシステムは、大学 1 年生を対象とした情報基礎科目において運用され、学生からは「候補ボタンが用意されていて調べやすかった」という肯定的意見や、「具体的な質問をしても内容が想定外だった」といった改善を求める意見が寄せられた。本研究ではシステムの概要が報告されており、授業や課題において疑問が生じた際にいつでも質問できる環境を提供しつつ、直接的な答えではなくヒントを提示することで、初学者の課題解決能力を高める可能性が示されている。今後の課題としては、システムの有用性をより大規模かつ定量的に評価することが求められる。

現代の労働市場において、従業員が長期間にわたり持続的に就労するためには、働きやすい職場環境の整備が不可欠である。日本では長時間労働や過剰な業務量、ワーク・ライフ・バランスの乱れが深刻な問題となっており、これらは従業員のストレス増加、モチベーション低下、さらには離職につながっている。また、労働人口の高齢化に伴う人材不足も深刻化しており、企業にとっては従業員の働きやすさを改善することが人材確保や生産性向上の観点から極めて重要である。

論文 [5] では、従業員満足度に影響を与える要因を明らかにすることを目的として、社員口コミデータの分析を実施している。具体的には、従業員が「働きやすさ」「仕事のやりがい」「会社への不満」などについて記載した口コミを分析対象とし、最先端の自然言語処理手法である BERTopic を用いて有効な情報を抽出するとともに、大規模言語モデルを活用したペルソナ導出により従業員の困りごとやモチベーション要因を可視化し、従業員満足度向上に向けた意思決定支援の方法を検討している。

本研究でこの論文を参考とした理由は、最先端の自然言語処理技術を実データに適用している点にあり、実験設計や分析手法が筆者の研究においても参考になると考えられるためである。分析の結果、口コミに共通して影響を与える要因を抽出し、ペルソナ設定を通じて従業員の特徴を明らかにすることに成功した。しかし一方で、満足度を具体的に高める施策の提言には至っていない。調査からは、スキルアップや人間関係の改善が満足度向上に寄与することが示唆されたが、施策立案そのものは現状の ChatGPT 等の生成 AI では困難であり、意思決定支援ツールとして部分的な有用性が確認されたにとどまっている。

本研究の目的は、大規模言語モデル ChatGPT を活用して、野球選手の個人特性に応じた練習支援システムを構築することである。具体的には、ChatGPT を Python アプリケーションに組み込み、ユーザが入力した 身長・体重・利き腕・長所・短所 といった情報に基づき、個別化された練習方法を自動的に生成・提示する仕組みを実装する。本システムにより、従来は指導者の経験や学習者自身の探索に依存していた練習方法の選定を、より簡便かつ個別化された形で提供することが可能となる。最終的には、野球に取り組む選手にとって効果的な練習支援を行い、パフォーマンス向上に資することを目指す。

3. システムの実装

3.1 システムの概略

本研究では、ChatGPT を活用した野球練習支援システムを開発した。開発言語は Python を使い、統合開発環境には Visual Studio Code (ver. 1.96.0) を使用した。実行環境は Windows 11 Home (ver. 23H2) 搭載の PC (LAPTOPM2JETU0H, Intel(R) Core(TM) i5-1035G1 CPU @ 1.00GHz, RAM 8.00GB) である。

システムの概要を以下に示す。ユーザは 身長・体重・利き腕・守備位置・長所・課題 の 6 項目を入力し、検索ボタンを押下すると、ChatGPT が入力情報をもとに個別化された練習方法を提案する。ChatGPT には事前に以下のプロンプトを与えている。

「あなたはプロの野球コーチです。この選手の長所を活かし、課題を克服するために具体的な野球の練習メニューを提案してください。例として『素振り 100 回』など、具体的な数値を含めて出力してください。最後に検索を容易にするためのキーワードを出力してください。」

身長・体重 を入力項目に含めた理由は、体格に応じて適切な練習方法や重点的に強化すべき要素が変化するためである。また、長所・課題 を入力させることで、選手の特

性を考慮した効率的かつ実践的な練習方法の提示が可能となる。さらに、長所や課題の記入に迷うユーザを想定し、入力例を赤字で表示することで操作性を高めている。

システム構成図を図 3.1 に示す。ユーザが入力したデータを ChatGPT が解析し、練習方法および関連キーワードを生成して出力する。本システムは Python アプリケーション上で ChatGPT API と連携させることで、実行環境上から直接 ChatGPT を利用可能としたものである。図 3.2 にアプリの基本画面、図 3.3 に検索結果の例を示す。

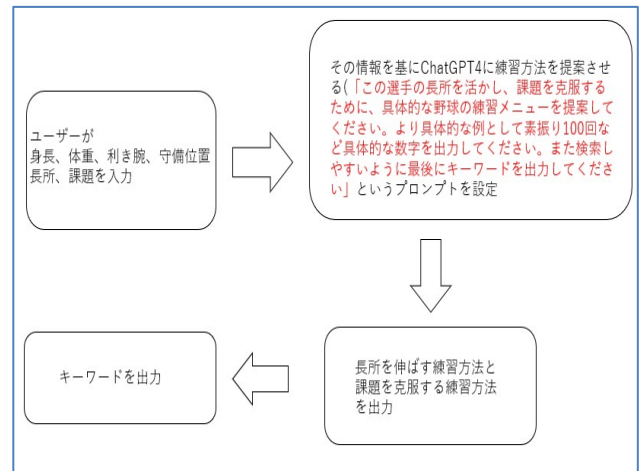


図 3.1 システム構成図

これらの項目を入力してください

長所の例: 肩が強い、走力がある、打撃が得意 など
課題の例: コントロールが悪い、変化球が苦手、守備範囲が狭い など

身長	
体重	
利き腕	
守備位置	
長所	
課題	

検索

図 3.2 システムの基本画面

3.2 システムを作成する前の準備

アプリケーションを開発するにあたり、まず Python 上で ChatGPT を利用可能にするための準備作業を行った。具体的には、OpenAI の公式サイトにアクセスしアカウントを作成した後、ダッシュボードから API キー を発行し

た。続いて、開発環境である Visual Studio Code (VSC) において OpenAI ライブラリをインストールし、Python から API を呼び出せるよう設定を行った。これにより、Python プログラム上で ChatGPT を活用可能な環境を整備した。

これらの項目を入力してください

長所の例: 肩が強い、走力がある、打撃が得意 など
課題の例: コントロールが悪い、変化球が苦手、守備範囲が狭い など

身長	165
体重	65
利き腕	右
守備位置	ショート
長所	肩が強い
課題	変化球が苦手

検索

長所: 肩が強い
1. 投球練習
強肩を活かすために、二塁送球とホーム送球の練習をメインに行います。始めは10メートルから始めて徐々に距離を広げていくことで、正確さと肩の持続力を両立する練習を行います。一日に最低100回の投球を目指しましょう。
2. インフィールド練習
ショートとしての守備範囲を広げるため、左右へのダイビングキャッチの練習やバックハンドキャッチの練習も行います。これも一日に100回を目指しましょう。

課題: 変化球が苦手
1. ティー練習
変化球への対応力を鍛えるために、まずはスタンドに固定したボールを打つティー練習から始めてみましょう。コントロールとスイングの基本を確認します。これを一日300回行います。
2. バッティングマシン練習
次に、バッティングマシンで変化球の打ち方を学びます。徐々にスピードや変化範囲を変化させていきます。ここでは一日200回のスイングを目指します。
3. ピッチャーとの対戦形式練習

図 3.3 検索結果

3.3 コードの説明

```
import tkinter as tk
import openai
import os
from tkinter import scrolledtext

# OpenAI APIキーの設定 (環境変数から取得)
openai.api_key = os.getenv("OPENAI_API_KEY")
```

図 3.4 コードの説明 1

図 3.4 に示すように、本システムでは tkinter をインポートすることで GUI を構築し、さらに openai をインポートすることで OpenAI の API を利用可能とした。openai.api_key は API キーを設定するためのプロパティであり、この設定により OpenAI のサービスへのアクセスが可能となる。

図 3.5 に示すコードは、Python で実装された関数 on_search() の一部であり、GUI アプリケーションにおい

```
def on_search():
    try:
        height = entry_height.get()
        weight = entry_weight.get()
        handedness = entry_handedness.get()
        position = entry_position.get()
        strength = entry_strength.get()
        weakness = entry_weakness.get()
```

てユーザ入力データを取得する役割を担う。on_search() 関数はユーザが「検索」ボタンをクリックした際に呼び出され、入力項目の読み取り処理を実行する。

図 3.5 コードの説明 2

```
# 入力のバリデーション
text_result.config(state=tk.NORMAL)
text_result.delete(1.0, tk.END) # テキストエリアをクリア

if not height.isdigit() or not weight.isdigit():
    text_result.insert(tk.END, "身長と体重には数字を入力してください。\\n")
    text_result.config(state=tk.DISABLED)
    return

if handedness not in ["右", "左"]:
    text_result.insert(tk.END, "利き腕には '右' か '左' を入力してください。\\n")
    text_result.config(state=tk.DISABLED)
    return
```

図 3.6 コードの説明 3

処理の冒頭には try: が記述されており、これは try-except ブロック の一部として、例外 (エラー) 発生時に適切な処理を行うための仕組みである。本システムでは、ユーザ入力の取得が失敗しやすい操作であることから、安全性を確保する目的でこの構造を採用している。

各入力値の取得は以下に行われる。

- height = entry_height.get(): ユーザが入力した 身長 を取得

- `weight = entry_weight.get()` : ユーザが入力した 体重 を取得
- `handedness = entry_handedness.get()` : ユーザが入力した 利き腕 を取得
- `position = entry_position.get()` : ユーザが入力した 守備位置 を取得 (例: ショート、ピッチャーなど)
- `strength = entry_strength.get()` : ユーザが入力した 長所 を取得
- `weakness = entry_weakness.get()` : ユーザが入力した 課題 を取得

以上により、本コードは 身長・体重・利き腕・守備位置・長所・課題の 6 項目を正確に取得し、後続の処理に渡すために実装されている。

図 3.6 に示すコードは、ユーザが入力した 身長 および 体重 の値が数値であるかを確認するためのバリデーション処理 である。

処理の流れは以下の通りである。

1. `text_result.config(state=tk.NORMAL)` により、テキストエリアを編集可能な状態に設定する。これにより、結果メッセージの挿入が可能となる。
2. `text_result.delete(1.0, tk.END)` により、テキストエリアの既存内容をすべて削除する。ここで 1.0 は最初の行の先頭、`tk.END` はテキストの終端を示す。
3. `height.isdigit()` および `weight.isdigit()` により、入力値が整数として妥当かを確認する。`isdigit()` は文字列が数字のみで構成されている場合に `True` を返し、それ以外の場合は `False` を返す。
4. `if not height.isdigit() or not weight.isdigit():` という条件文により、いずれか一方でも数値でない場合は条件が成立する。この際、`label_result.config(text="身長と体重には数字を入力してください。")` が実行され、エラーメッセージをラベルに表示する。`config()` メソッドを利用することで、ラベル内容を動的に更新している。
5. エラーメッセージ表示後、`return` により関数の処理を終了する。これにより、不正な入力が行われた場合は後続処理が実行されない仕組みとなっている。

本処理により、ユーザ入力の妥当性を保証し、誤入力による後続処理の不具合を防止している。

図 3.7 に示すコードは、ChatGPT API を利用して具体的な練習方法を提案する処理である。`prompt` には、ユー

ザが入力した 身長・体重・利き腕・守備位置・長所・課題の 6 項目とともに、次の指示文を与えている。

```
# ChatGPT APIを使用して具体的な練習方法を提案
prompt = f"身長: {height}cm, 体重: {weight}kg, 利き腕: {handedness}, 守備位置: {position}\n \
f"長所: {strength}, 課題: {weakness}\n \
"この選手の長所を活かし、課題を克服するために、具体的な野球の練習メニューを提案してください。"

response = openai.ChatCompletion.create(
    model="gpt-4", # gpt-4を指定
    messages=[
        {"role": "system", "content": "あなたはプロの野球コーチです。"},
        {"role": "user", "content": prompt}
    ],
    max_tokens=1500
)
```

図 3.7 コードの説明

「この選手の長所を活かし、課題を克服するために、具体的な野球の練習メニューを提案してください。例として『素振り 100 回』のように具体的な数値を含めて出力してください。また、検索を容易にするために最後にキーワードを出力してください。」

これにより、ChatGPT は入力情報に基づき、個人特性に適合した練習方法を生成するよう誘導される。

コード中の `openai.ChatCompletion.create` は ChatGPT API を呼び出し、指定したモデルとメッセージに基づいてテキストを生成する関数である。戻り値の `response` には AI が生成した応答データが格納される。本研究では GPT-4 モデル を指定しており、その理由は GPT-4 が高精度かつ高度な応答を生成でき、特定のタスクに適した回答を導出できるためである。

また、`messages` 引数では ChatGPT に渡すメッセージをリスト形式で指定している。ここで `role: system` には「あなたはプロの野球コーチです」と記述し、AI にプロ野球コーチとして振る舞わせるように設定している。一方、`role: user` には、ユーザが入力した選手情報を含むプロンプトを渡している。最後に `max_tokens` を指定し、応答の長さを制御している。本システムでは 1500 トークン以内に収まるよう設定することで、十分な情報量を確保しつつ応答の冗長化を防いでいる。

```
# レスポンスの取得と処理
training_plan = response['choices'][0]['message']['content'].strip()

# ChatGPTの出力をテキストエリアに表示
text_result.insert(tk.END, training_plan)
text_result.config(state=tk.DISABLED)

except Exception as e:
    text_result.config(state=tk.NORMAL)
    text_result.insert(tk.END, f"エラーが発生しました: {e}")
    text_result.config(state=tk.DISABLED)
```

図 3.8 コードの説明

図 3.8 に示すコードは、ChatGPT API から取得したレスポンスを処理し、テキストエリアに表示する処理およびエラー処理を実装した部分である。

まず、`training_plan = response` により API から返されたレスポンスを抽出し、練習メニューとして利用する。続いて、`text_result.insert(tk.END, training_plan)` によりテキストエリア末尾に練習メニューを挿入し、ユーザーに提示する。さらに、`text_result.config(state=tk.DISABLED)` によってテキストエリアを読み取り専用を設定し、ユーザーによる誤編集を防止している。

図 3.9 に示すコードは、GUI の基盤構築からフォント設定および入力補助ラベルの追加までを行う部分である。具体的には以下の処理を実装している。

1. `tk.Tk()` により新規ウィンドウを生成し、`root.title(...)` にてタイトルを「ChatGPT4 を利用した野球選手のための練習方法提案システムの試作」に設定する。
2. `root.geometry()` でウィンドウのサイズを幅×高さの形式で指定し、表示の見やすさを確保する。
3. フォント設定として、通常ラベル用に Arial 14pt (`label_font`)、太字ラベル用に Arial 14pt Bold (`bold_label_font`)、ボタン用に Arial 14pt (`button_font`) を定義している。
4. `tk.Label()` によりテキストを表示するラベルを生成し、`font` オプションでフォントスタイルを指定する。`pack()` メソッドでは上下に 10 ピクセルの余白を追加し、要素間のバランスを調整する。
5. `example_text` には長所と課題の入力例を格納し、`label_examples = tk.Label()` で例示を表示するラベルを作

成している。この際、フォントやテキストカラーなどのスタイルも同時に設定している。

これらの処理により、ユーザーにとって見やすく操作しやすい GUI を実現し、入力支援の利便性を高めている。

```
# GUIの設定
root = tk.Tk()
root.title("Chatgpt4を利用した野球選手のための練習方法提案システムの試作")

# ウィンドウサイズを設定
root.geometry("800x700") # 高さを少し広げる

# フォント設定
label_font = ("Arial", 14)
bold_label_font = ("Arial", 14, "bold") # 太字フォントを定義
button_font = ("Arial", 14)

# 「これらの項目を入力してください」のラベルを追加（太字に変更）
label_instruction = tk.Label(root, text="これらの項目を入力してください", font=bold_label_font)
label_instruction.pack(pady=10)

# 長所と課題の例のラベルを追加（文字色を赤に変更）
example_text = "長所の例: 肩が強い、走力がある、打撃が得意 など\n\n課題の例: コントロールが悪い、変化球が苦手、守備範囲が狭い など"
label_examples = tk.Label(root, text=example_text, font=("Arial", 12), fg="red", justify="left", wraplength=750)
label_examples.pack(pady=5)
```

図 3.9 コードの説明

図 3.10 に示すコードは、Python の tkinter ライブラリを用いて、ユーザー入力用のラベルおよび入力フィールドを生成する処理である。まず、`frame_inputs` は入力フォーム全体をまとめるためのフレームであり、`tk.Frame(root)` によりメインウィンドウ内に作成され、`pack()` によって配置される。その後、身長・体重・利き腕・守備位置・長所・課題の 6 項目について、それぞれ `tk.Label` によりラベルを作成し、`tk.Entry` により入力フィールドを生成している。

図 3.11 のコードは、検索ボタンと結果表示用のスクロール付きテキストエリアを生成する処理である。`tk.Button` を用いて検索ボタンを作成し、テキストを「検索」と指定している。ボタンは `on_search` 関数と連動し、`font=button_font` でフォントを指定、`width=10` および `height=2` でサイズを設定し、`pack(pady=20)` によりウィンドウ内に配置される。

続いて、`scrolledtext.ScrolledText` を利用してスクロール機能付きのテキストウィジェットを生成し、長文の出力であっても容易に閲覧できるようにしている。最後に、`root.mainloop()` を実行することで GUI ウィンドウを維持し、ユーザー操作を可能にしている。

```

# 入力項目のラベルとエントリ
frame_inputs = tk.Frame(root)
frame_inputs.pack()

label_height = tk.Label(frame_inputs, text="身長", font=label_font)
label_height.grid(row=0, column=0)
entry_height = tk.Entry(frame_inputs, font=label_font)
entry_height.grid(row=0, column=1)

label_weight = tk.Label(frame_inputs, text="体重", font=label_font)
label_weight.grid(row=1, column=0)
entry_weight = tk.Entry(frame_inputs, font=label_font)
entry_weight.grid(row=1, column=1)

label_handedness = tk.Label(frame_inputs, text="利き腕", font=label_font)
label_handedness.grid(row=2, column=0)
entry_handedness = tk.Entry(frame_inputs, font=label_font)
entry_handedness.grid(row=2, column=1)

label_position = tk.Label(frame_inputs, text="守備位置", font=label_font)
label_position.grid(row=3, column=0)
entry_position = tk.Entry(frame_inputs, font=label_font)
entry_position.grid(row=3, column=1)

label_strength = tk.Label(frame_inputs, text="長所", font=label_font)
label_strength.grid(row=4, column=0)
entry_strength = tk.Entry(frame_inputs, font=label_font)
entry_strength.grid(row=4, column=1)

label_weakness = tk.Label(frame_inputs, text="課題", font=label_font)
label_weakness.grid(row=5, column=0)
entry_weakness = tk.Entry(frame_inputs, font=label_font)
entry_weakness.grid(row=5, column=1)

```

図 3.10 コードの説明

```

# 検索ボタン
button_search = tk.Button(root, text="検索", command=on_search, font=button_font, width=10, height=2)
button_search.pack(pady=20)

# スクロールテキストエリアを作成
text_result = scrolledtext.ScrolledText(root, wrap=tk.WORD, width=80, height=20, font=("Arial", 12))
text_result.pack(pady=10)
text_result.config(state=tk.DISABLED)

root.mainloop()

```

図 3.11 コードの説明

4. 評価実験

4.1 実験設定

本研究で試作したシステムの有用性を検証するため、野球未経験者 5 名および野球経験者 5 名を対象として評価実験を実施した。被験者には本システムを一定時間利用させ、その操作性および生成される出力について評価を求めた。

評価は 5 段階リッカート尺度を用いて行い、1 を最低評価、5 を最高評価とした。評価項目は以下の 3 点である。

1. UI の評価：画面構成や操作性に関する使いやすさの評価

2. 応答時間：入力から応答が返されるまでの処理速度に関する評価

3. 応答の質：提示される練習方法の具体性・妥当性に関する評価

これらの基準に基づき、未経験者と経験者それぞれにおける評価傾向の差異を明らかにすることを目的とした。

4.2 実験結果

実験の結果を表 4.1、表 4.2 に示す。

表 4.1 野球未経験者の評価

野球経験者	UIの評価	応答時間	応答の質
Kさん	5	5	3
Tさん	4	5	2
Mさん	3	4	2
Kさん	3	3	2
Hさん	5	5	4

表 4.2 野球経験者の評価

野球未経験者	UIの評価	応答時間	応答の質
Kさん	4	3	5
Mさん	5	4	5
Kさん	5	3	4
Tさん	3	4	3
Yさん	4	5	4

表 4.1 に野球未経験者の評価結果を、表 4.2 に野球経験者の評価結果を示す。両者を比較すると、未経験者（表 4.1）では UI の平均評価が 4.2、応答時間が 3.8、応答の質が 4.2 であり、総じて高評価を得た。一方、経験者（表 4.2）では UI が 4.0、応答時間が 4.4 と一定の評価を得たものの、応答の質は 2.6 にとどまり、低評価となった。

未経験者からは「必要十分な応答が得られた」「練習方法が分かりやすい」「具体的な数値を用いた説明で理解しやすい」「応答時間が早く的確であった」など、肯定的な意見が多く寄せられた。これに対し、経験者からは「提案内容が想定通りで新規性に乏しい」「練習内容が過剰で実

現性に欠ける」「表現が分かりづらい」「具体的ではあるが現実性に欠ける」といった否定的意見が見られた。

以上の結果から、本システムは未経験者に対しては有用である一方、経験者にとっては内容の妥当性や実効性に課題が残ることが示された。

5. おわりに

本研究では、ChatGPT を活用し、野球選手向けの練習方法提案システムを試作した。本システムは、選手の身長・体重・利き腕・守備位置・長所・課題の6項目を入力とし、それに基づいて ChatGPT が長所を伸ばし課題を克服するための具体的な練習方法を提示するものである。

評価実験では、野球未経験者5名と経験者5名を対象に、UIの評価・応答時間・応答の質の3項目について5段階評価を実施した。その結果、UIと応答時間は両群で高評価を得た一方、応答の質については未経験者からは有用性が認められたものの、経験者に対しては十分ではないことが明らかとなった。これは、経験者がより専門的かつ実践的な練習方法を求める傾向にある一方で、現行システムではその要求水準を満たす出力を生成できなかったためと考えられる。

以上の結果から、本システムは野球未経験者にとって実用的かつ分かりやすい練習方法を提供できる一方、経験者に対しては応答の質をさらに向上させる必要があるといえる。今後の展望としては、プロンプト設計の改善や専門知識の統合により応答内容を高度化し、経験者にとっても有用性の高いシステムを目指す。また、野球未経験者に対しては、具体的かつ実践的な練習方法を提示することで競技参加の動機づけを支援できる点で、本研究の意義は大きい。

本研究の成果として、①未経験者に有用な練習支援を提供するシステムを構築したこと、②UIと応答速度に関して高い評価を得たこと、③ChatGPT-4を活用したスポーツ領域でのトレーニング支援の新たな可能性を提示したことが挙げられる。特に、野球未経験者に対しては練習方法の理解を促進し、競技参加を後押しするツールとして一定の貢献があったと評価できる。

参考文献

- [1] ChatGPT:(<https://chatgpt.com/c/672bada5-f034-8012-a9cf-b71c2d89b9e>)(2024年11月7日)
- [2] 高野 海斗, 中川 慧, 藤本 勇吾, "ChatGPT を活用した運用報告書の市況コメントの自動生成", 人工知能学会第二種研究会資料, pp.61~67, (2023年)

- [3] 木村 祐介, 杉本 徹, "単語分散表現を用いた対話型観光地推薦システム", ファジィシステムシンポジウム講演論文集 (CD-ROM)(ファジィシステムシンポジウム (CD-ROM)) 巻:39thROMBUNNO.2D3-1, pp1~4, 発行年(2023年)
- [4] 八木 俊磨, 漆原 宏丞, 鈴木 達也, 島袋 舞子, 荒木 千秋, 兼定 進, "生成系 AI を活用した対話型学習支援システムの開発", 情報教育シンポジウム論文集, pp225~230 (2024-08-03)
- [5] 神山 直也, 高橋 大志, "自然言語処理技術および大規模言語モデルを用いた従業員の労働環境分析", 人工知能学会第二種研究会資料 2024 巻 BI-024 号, pp16~23, (2024年)
- [6] 野呂 悠斗, 砂山 渡, 服部 峻, "ChatGPT を用いた対話相手の好感度を高められるコミュニケーション支援システム", 2024 年度人工知能学会全国大会, pp1~4, (2024年)
- [7] 谷口 拓紀, 砂山 渡, 服部 峻, "ChatGPT を用いた 2 つの事柄の比較支援システム", 2024 年度人工知能学会全国大会論文集, pp1~4, (2024年)
- [8] 桑田 喜隆, 石坂 徹, 政谷 好伸, 横山 重俊, 浜元 信州, 谷沢 智史, "プログラミング演習支援スタイルの生成系 AI を活用した検証方法に関する提案", 人工知能学会第二種研究会資料, 2023 巻 KSN-034 号, pp35~40, (2024年)