

農業情報学会 2024 年度年次大会個別口頭発表
高崎健康福祉大学
2024 年 5 月 26 日

大規模言語モデルと連携した 農場ディープデータの解析環境の 設計と労働時間分析への適用

¹ 農研機構 北海道農業研究センター

² もうたい農場

○佐藤 正衛¹ 馬渡 智昭²

- ① はじめに
- ② 開発する解析環境の概要と特徴
- ③ 労働時間分析への適用
- ④ おわりに
- ⑤ 付録: 構築した解析環境を用いた分析ワークフローのイメージ

- 農場ディープデータの蓄積が加速化
- 個別ニーズにそってデータを経営意思決定支援情報として活用
- そこで、以下の構成で解析環境を構築した
 - 前提：大規模言語モデル LLM のサービスを連携利用
 - テキストエディタ
 - API クライアントプログラム
 - R 言語とその関連パッケージ
 - Emacs の入力支援機能，拡張性を利用
- その結果，関数作成を LLM で支援するアプローチで，情報セキュリティや回答の正確性の確保等のシステム要件を一定程度，満たすことが可能

デジタル農業技術の進展

- 農場の生産プロセスから発生する多種多様なデータの蓄積を加速
 - しかし、これらのデータが十分に活用されていない
 - その理由は、提供されている分析サービスと農業経営者の個別ニーズとの間に乖離
- 生成 AI 技術が急速に発展
 - データ分析等利用への期待
- → ニーズ乖離の解消、技術革新の成果を経営の発展に繋げる

デジタル技術革新とリスク

- データやノウハウの流出，多様性の喪失等のリスク (南石, 2022)
- AI の開発，運用，利用に関わる事業者に対するガイドライン策定 (総務省，経済産業省, 2024)
 - 利用者：適正利用や人間の判断を介在させた運用

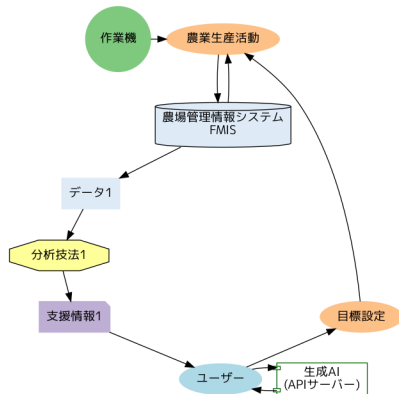


図 1: 営農支援事例から描かれた生産経営情報と PDCA サイクル (2023 年度 JSAI 生産・経営情報部会 OS プレゼン資料 (佐藤 2023) より引用。)

デジタル時代の農業経営分析

- デジタル技術リスクへの指摘をふまえて、
- LLM を効率的かつ効果的に利用する **解析環境を提案** し、
- 蓄積する農場データの分析への大規模言語モデル LLM の利用可能性を検討

- ① はじめに
- ② 開発する解析環境の概要と特徴
- ③ 労働時間分析への適用
- ④ おわりに
- ⑤ 付録: 構築した解析環境を用いた分析ワークフローのイメージ

Definition (農場ディープデータとは)

「特定の農場や営農方式の個別の状況に依存し、その耕地生態系内の複雑な因果関係に基づいて定義されるデータである。このデータは、土壌・気象条件、作物の種類とその生育状況、病害虫の発生状況、農作業の詳細かつ特定の農業環境に特有の情報を含んでいる。このディープデータを深く分析し、モデルを常に精査して利活用していくことで、自農場のビジネス価値を大幅に高める。」

西岡 (2020) を参照し、農場ディープデータを定義した。

方針

- 解そのものを LLM に求めるのではなく、LLM を使って分析ワークフローや解法の生成を組織内で内製化，蓄積する

要点

- (a) 情報セキュリティの確保
- (b) 大規模言語モデルの回答の正確性
- (c) 再利用可能性の確保
- (d) ディープデータ分析の柔軟性
- (e) 技術革新に対応するシステムの拡張性

方針

- 解そのものを LLM に求めるのではなく、LLM を使って分析ワークフローや解法の生成を組織内で内製化，蓄積する

要点

- (a) 情報セキュリティの確保
- (b) 大規模言語モデルの回答の正確性
- (c) 再利用可能性の確保
- (d) ディープデータ分析の柔軟性
- (e) 技術革新に対応するシステムの拡張性

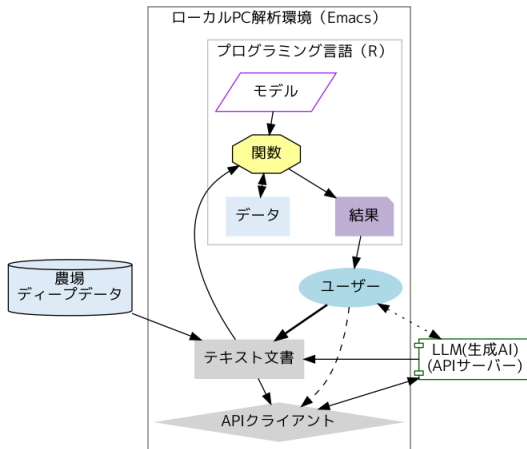


図 2: LLM と連携した農場ディープデータの解析

表 1: 解析環境の構築に用いたソフトウェア等

種類	ソフトウェア等名	内容
テキストエディタ	Emacs	インタフェースの基盤
API クライアント	R+シェルスクリプト	LLM へリクエスト送信
API サーバー	OpenAI gpt-3.5-turbo	LLM。リクエストからテキスト生成
開発言語	R, Lisp	プログラミング
入力支援ツール	ESS, transient, yasnipet	入力やコーディング支援

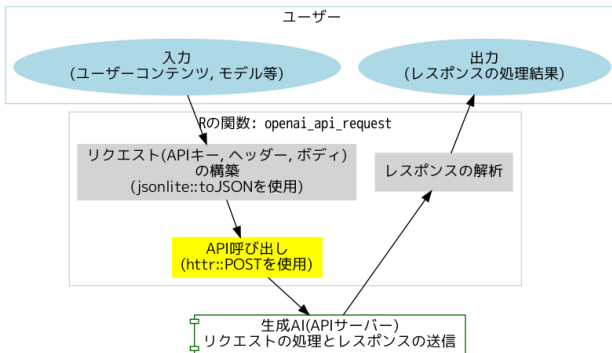


図 3: api リクエスト関数の引数に質問内容, 利用モデルを与えて, AI からレスポンス (回答) を取得する。

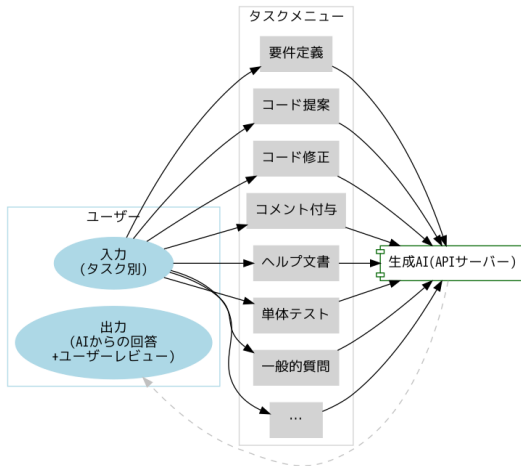


図 4: タスク別に事前に定義したプロンプトを API で AI へ送る。得られた回答をユーザーが評価し利用する。

- ① はじめに
- ② 開発する解析環境の概要と特徴
- ③ 労働時間分析への適用
- ④ おわりに
- ⑤ 付録: 構築した解析環境を用いた分析ワークフローのイメージ

ディープデータによる時間分析

- 内容：
 - 時間当たりの利益は、経営者の目標値として重要
 - しかし、労働時間、作業時間、機械稼働等の時間分析が実施されている事例は少ない
 - ディープデータ解析の時間、価値を評価
- 方法：
 - 日誌記録，作業ログ，農機端末内データのデータ収集
 - 分析用の関数を作成
 - 一連の分析フローを実行し，システム要件項目ごとに評価

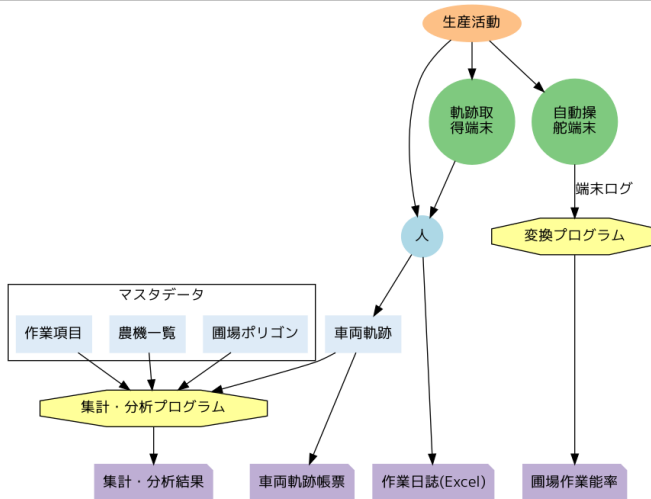


図 5: 収集データの分析フローの概要

(a) 情報セキュリティ

- 方針：ローカル PC 環境で，作成した関数を用いて分析
- → 経営データは通信しなくてよい。秘密情報を LLM に送らなくてよい。
- ただし，実際のファイルを指定したり，サンプルデータがあるほうが関数構築が効率的な場合あり
- テキストエディタ上に営業秘密情報を記述しない等，運用面の注意は必要

(b) LLM の回答の正確性

- 利用 LLM gpt-3.5-turbo：プログラムのコーディング機能の水準は良好
- タスク別のプロンプトを作成 → 回答精度が向上
- 要件定義は，構造化したテンプレートを準備。スニペット機能で入力 of 簡素化可能。→ 回答精度が向上
- 利用言語の特有の用語，利用パッケージ名等を用いる → 回答精度が高まる傾向

(a) 情報セキュリティ

- 方針：ローカル PC 環境で、作成した関数を用いて分析
- → 経営データは通信しなくてよい。秘密情報を LLM に送らなくてよい。
- ただし、実際のファイルを指定したり、サンプルデータがあるほうが関数構築が効率的な場合あり
- テキストエディタ上に営業秘密情報を記述しない等、運用面の注意は必要

(b) LLM の回答の正確性

- 利用 LLM gpt-3.5-turbo：プログラムのコーディング機能の水準は良好
- タスク別のプロンプトを作成 → 回答精度が向上
- 要件定義は、構造化したテンプレートを準備。スニペット機能で入力 of 簡素化可能。→ 回答精度が向上
- 利用言語の特有の用語、利用パッケージ名等を用いる → 回答精度が高まる傾向

(c) 再利用可能性の確保

- 作成した関数は、Rの開発ワークフローに従ってパッケージ化。
- → ライブラリ間の相互依存関係やドキュメントの不備等が防止・低減効果。一定水準の品質のソフトウェア。

(d) ディープデータ分析の柔軟性

- 畑作経営のデータ：多種多様な時空間データ → 有用なパッケージの調査等は今後の課題。

(c) 再利用可能性の確保

- 作成した関数は、Rの開発ワークフローに従ってパッケージ化。
- → ライブラリ間の相互依存関係やドキュメントの不備等が防止・低減効果。一定水準の品質のソフトウェア。

(d) ディープデータ分析の柔軟性

- 畑作経営のデータ：多種多様な時空間データ → 有用なパッケージの調査等は今後の課題。

(e) システムの拡張性

- 比較的容易に実施可能
- ① 他の代替 LLM を利用
- ② タスク種類を追加
- ③ タスク別プロンプトを改善
- ④ 各種テンプレートの追加
- ⑤ ワークフローのコマンド化
- Emacs 自体を Lisp 言語で柔軟に機能拡張可能

その他

- 独自 MS-Excel ファイルで管理されるデータ多い
- Excel ファイル → 複雑な構造であることが多く、読み込み方法に工夫
- バイナリデータ → メーカーに仕様確認、変換対応を依頼

(e) システムの拡張性

- 比較的容易に実施可能
- ① 他の代替 LLM を利用
- ② タスク種類を追加
- ③ タスク別プロンプトを改善
- ④ 各種テンプレートの追加
- ⑤ ワークフローのコマンド化
- Emacs 自体を Lisp 言語で柔軟に機能拡張可能

その他

- 独自 MS-Excel ファイルで管理されるデータ多い
- Excel ファイル → 複雑な構造であることが多く、読み込み方法に工夫
- バイナリデータ → メーカーに仕様確認、変換対応を依頼

- ① はじめに
- ② 開発する解析環境の概要と特徴
- ③ 労働時間分析への適用
- ④ おわりに
- ⑤ 付録: 構築した解析環境を用いた分析ワークフローのイメージ

農場ディープデータの解析環境

- 到達点
 - LLM に経営分析の解ではなく解法を求めることとした
 - LLM 連携の解析環境の基本的構成要素を選択しシステム化した
 - 営農ディープデータを用いて労働時間の分析に取り組み、開発システムの評価を実施した
- 今後の課題
 - 労働時間分析を進め営農支援情報としての有効利用に取り組む
 - 比較検討：RStudio, Visual Studio Code, JupyterNotebook 等

南石晃明編 (2022) デジタル・ゲノム革命時代の農業イノベーション, 農林統計出版.

西岡靖之 (2020) ディープデータを介した製造業のデジタルトランスフォーメーション, 「2020 年版ものづくり白書」, 経済産業省, 80-81.

総務省, 経済産業省 (2024) AI 事業者ガイドライン (第 1.0 版).

- ① はじめに
- ② 開発する解析環境の概要と特徴
- ③ 労働時間分析への適用
- ④ おわりに
- ⑤ 付録: 構築した解析環境を用いた分析ワークフローのイメージ

実施するタスクの手順

ほとんどのタスクは transient ツールを用いたコマンド中心の作業

- ① テキストエディタで R を開始
 - Emacs + ESS の利用
- ② テンプレートと概要説明から要件定義
- ③ 実装する分析用関数を AI で提案，修正
- ④ ドキュメント作成：関数作成後にコメントやヘルプ文書を作成
- ⑤ パッケージ化

手順

- コマンドで要件定義のテンプレートを表示
- 実装したい機能の概要を記述
- 定義 (案) を AI で作成
- 案を修正して要件定義を完成

```

2+ example
6 ## 例題
7 ■ 要件定義
8 ## 以下のテンプレートを使用して,
9 ## 1. データの読み込み: ./data/ディレクトリから気象のcsvファイルを読み込む関数を実装す
10 ## 2. 読み込んだ気象データから月ごとの平均気温, 最高気温, 最低気温を計算する関数を実装
11 ## 3. 月ごとの気温データをグラフにプロットする関数を実装する
12 ■ 関数名: myfunction
13 ○ 関数の目的:...
14 ○ 入力:...
15 ○ 出力:...
16 ○ 動作:...
17 ○ 例外処理:...
18 ○ 副作用:...
19 ○ パフォーマンス:...
20 ○ 制約:...
21 ○ 入力例 (関数の使用例) :...
22 ○ 出力例 (出力結果の形式の具体例) :...
--かな: U:*** presen_example.R Bot L6 [2] [(ESS[R] [R] [+])]
修正 翻訳 プログラミング Rプログラミング AI org-ad
ca 学術的文書 te →英語 pp Pythonへ変換 rd 要件定義 Ac チャ
cb ビジネス文書 tj →日本語 pr Rへ変換 rp 要件からコード提案 Ar 選択
cg 一般 rf コードの内容説明
rc 関数にコメント付与
ro roxygen2
rt 単体テスト
rr リファクタリング

```

図 6: 実装したい機能の説明と要件定義テンプレートを表示して, [rd: 要件定義] コマンドを実行

手順

- 要件定義を選択
- 実装したい関数 (案) を AI で作成
- 案を修正して関数を完成

```
2+ example
60 関数名: read_weather_data
61 ## 関数の目的:
62 ## データの読み込みを行い、気象データを返す。
63
64 ## 入力:
65 ## - 入力1: directory_path (character), データのディレクトリパス
66 ## - 入力2: file_name (character), ファイル名
67
68 ## 出力:
69 ## - 出力1: data.frame, 読み込んだ気象データ
70
71 ## 動作:
72 ## 1. 指定されたディレクトリから指定されたファイルを読み込む。
73
74 read_weather_data <- function(directory_path, file_name) {
75   data <- read.csv(paste0(directory_path, "/", file_name))
76   return(data)
77 }
78
79 関数名: calculate_monthly_weather_stats...
105 関数名: plot_monthly_weather_data...
```

図 7: 要件定義を選択して [rp] コマンドを実行。案を修正し完成

A3. ヘルプドキュメントの作成

手順

- 適宜コメント加筆して関数を完成
- 関数のヘルプ文書を作成
- 使用例なども追加

```
82 roxygen2
83 read_weather_data
84 roxygen2
85 ##' Function to read weather data from a CSV file located in the specified
86 ##' and with the given file name.
87 ##'
88 ##' @title Read Weather Data
89 ##' @author [Your Name]
90 ##' @param directory_path Path to the directory where the CSV file is located
91 ##' @param file_name Name of the CSV file to be read
92 ##' @return The data read from the CSV file
93 ##' @examples
94 ##' read_weather_data("path/to/directory", "weather_data.csv")
95
96 function
97 read_weather_data <- function(directory_path, file_name) {
98   # Read the CSV file by combining the directory path and file name
99   data <- read.csv(paste0(directory_path, "/", file_name))
100   return(data)
101 }
102
103
104 関数名: calculate_monthly_weather_stats...
105 関数名: plot_monthly_weather_data...
```

図 8: 関数を選択して [ro] コマンドを実行。ドキュメント案を修正し完成

ポイント

- 既存のツール (R devtools) を利用してチェック
 - 既成ツール群 (devtools, testthat, roxygen2, testthat, usethis) を利用
- 再利用性, 共有性, メンテナンス性が向上