AI-Driven Virtual Assistance Interface for Excel-Based Economic Model

[EE494]

Shubhram Pandey[1], Rajdeep Kaur[1], Siguroli Teitsson[2], Bill Malcolm[2], Pankaj Rai[1], Barinder Singh[3], Sven Klijn[4]

[1]Pharmacoevidence, Mohali, India, [2]Bristol Myers Squibb, Uxbridge, UK, [3]Pharmacoevidence, London, UK, [4]Bristol Myers Squibb, Lawrence Township, NJ, USA

Introduction

- Excel-based economic models are extensively used for cost-effectiveness analysis that plays a crucial role in decision-making processes. This research work presents an AI-Driven virtual assistance interface for Excel-based economic models.
- The AI assistant facilitates performing different operations on Excel-based economic models such as data retrieval and updates. An essential characteristic of the AI-based virtual assistant is its capability to facilitate multi-parameter changes using a template mapping approach.
- The assistant has an intuitive interface with a dropdown menu for the selection of model parameters, enhancing user experience. This feature eliminates the need for manual input of parameter names when writing the natural language queries and significantly streamlines the overall user interaction process.

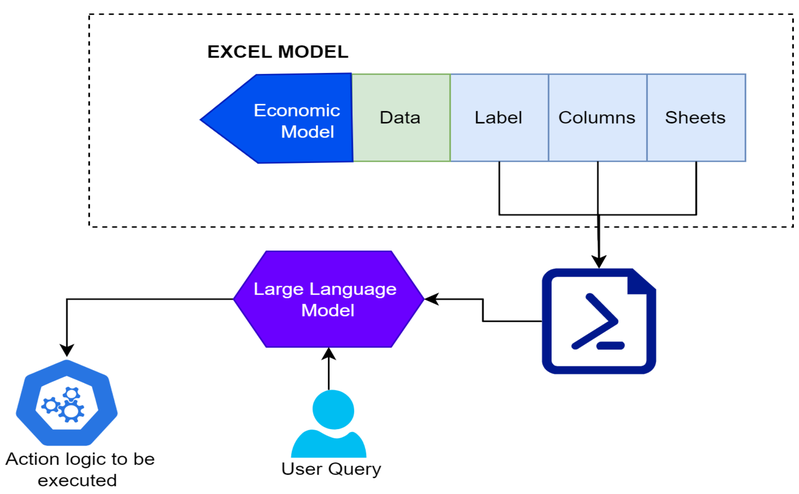
Objective

- This study aims to develop a virtual assistant interface specifically designed for bespoke Excel-based economic models. This interface will leverage the capabilities of a large language model (LLM), Claude-3, to perform various tasks within the model and customize it for different markets

What this study adds

- Flexibility: Enables advanced customization and market-specific adaptations through intelligent multi-parameter update capabilities
- Security: Develops a strong and efficient strategy for extracting and updating data in Excel-based economic models using Generative AI while maintaining confidentiality at all stages (Figure 1)
- Efficiency: Significantly reduces the overhead associated with manual data retrieval and updates in complex CE models.
- Usability: The platform has been designed to support complex tasks, allowing those without expert knowledge of the Excel models to interact with the models

Figure 1. Ensuring Data Security



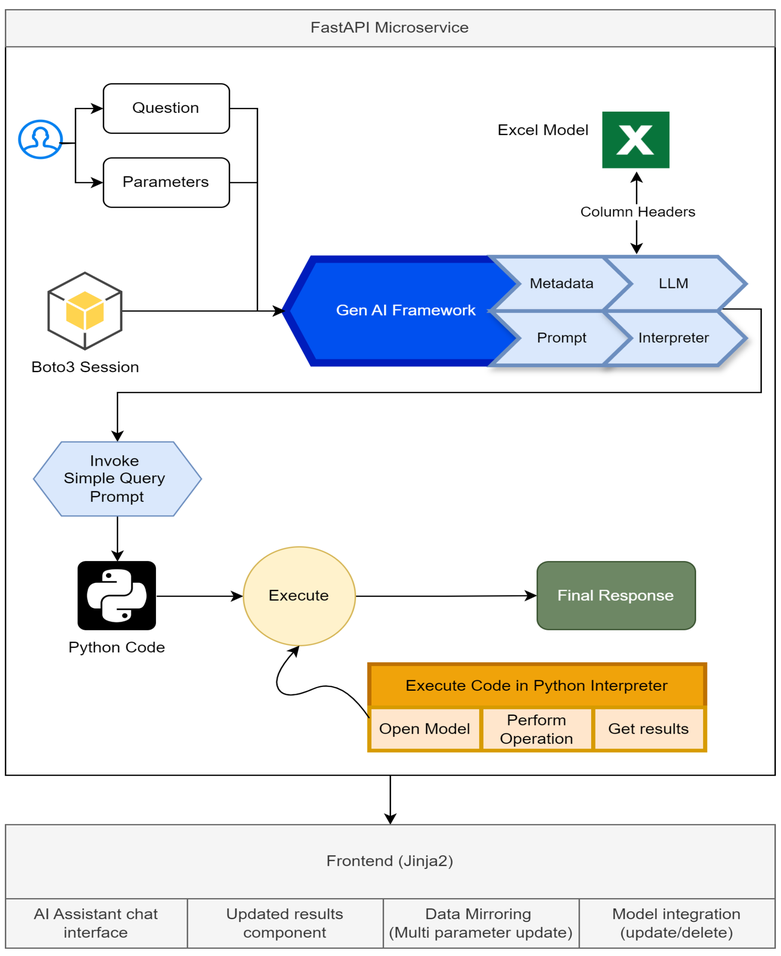
Methods

System Architecture Overview

The system employs a multi-layered architecture with an advanced framework for generative AI, using capabilities of the Claude 3 Opus model (Figure 2)

- The LLM is integrated through the LangChain framework, thus allowing for a dynamic generation of Python code based on the user input and model requirements
- A major application of this approach is the dynamic, real-time generation of Python code that is tailored to specific Excel functions and user-specific queries
- Users can input natural language prompts and select model parameters from a dynamically generated dropdown menu
- The LLM processes these inputs with system prompts and Excel column headers, generating logic codes without transmitting Excel data, ensuring privacy
- The produced code is run in a controlled interpreter environment, ensuring sensitive information is not leaked to systems outside the controlled Excel model interaction
- The frontend part of the system, built with Jinja2 templates, is an intuitive and responsive way to access complex Excel-based economic models. This accessible layer is cohesively linked to a robust backend by FastAPI, a modern and high-performance web framework especially designed for developing APIs based on Python

Figure 2. Framework



LLM Integration and configuration

- The interface was powered by Anthropic's Claude 3 Opus model, which was integrated using AWS Bedrock. Bedrock provides a secure and scalable platform that enhances data processing activities required for economic modelling
- The model version "anthropic.claude-3-opus-20240229-v1:0" was selected due to its superior code generation performance, at the time of development of the framework, compared to previous versions
- A low temperature setting of 0.1 was implemented to ensure consistent and precise outcomes. These parameter settings were important for providing consistent results and reducing errors in code development, and are critical for economic modelling

Methods

- The boto3_bedrock client was employed in the configuration, which was optimised with precise delay settings to efficiently manage the complex calculations associated with CE model manipulations (Figure 3)

Figure 3. Configuration Snippet

```
boto3Session = boto3.Session()
config = Config(read_timeout=1500)
boto3_bedrock = boto3Session.client("bedrock-runtime", region_name='us-west-2', config=config)
claudeV2_llm = BedrockChat(
    model_id="anthropic.claude-3-opus-20240229-v1:0",
    streaming=True,
    client=boto3_bedrock,
    model_kwargs={"temperature": 0.1}
)
```

Dynamic Code Generation and Execution

The interface streamlined the processes of manipulating and updating economic models by enabling the creation of dynamic code via inputs provided in natural language

1. The user can enter a structured query, for example: "Update Time Horizon, willingness to pay by 3 and 50000"
2. To generate the Python code, the LLM was provided with the context information, which includes information regarding the Excel model structure and available operations
3. The LLM generated the Python code as per the user's query and context information
4. The generated code executed in a controlled Python environment, which includes every essential library (such as xlwings)
5. The code ensures that the model updated to reflect the most recent changes by invoking app.api.Calculate() after each modification

Prompt framework

The interface used a structured approach to prompt the LLM for generating Python code to work with an Excel-based economic model. The prompt framework employed the following sections:

Context Section: included a context section marked with XML-like tags, background details, including the Excel model structure.

User Request and Labels: It included the user's specific query along with relevant column headers or labels from the Excel model (e.g., 'Labels' in column C and 'Values' in column E).

Constraints and Error Handling: Guidelines for managing errors, try-except blocks, instructions to handle incorrect or missing labels.

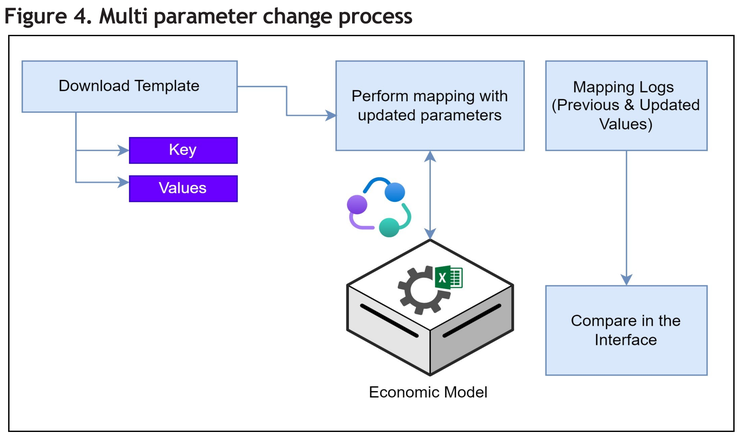
Code Style Rules: Specific rules for code style, rounding of results, and handling different data types.

Dynamic Input Handling: Included guidelines specify how user input dynamically integrates with model-specific data into the prompt. The user question is inserted into the {question} placeholder, relevant Excel labels are inserted into the {labels_list} placeholder, and the {context} is filled with a comprehensive string, including instructions and examples, and the path to Excel-based economic models.

Template-Based Mirroring

- The template-based mirroring interface allows users to change multiple parameters of an Excel-based economic model simultaneously by uploading the new values in the model using a template (Figure 4). This procedure facilitates the management of large-scale bulk updates in a more organized and efficient manner using the following steps:

Figure 4. Multi parameter change process



Download Template: The user has the option to download a formatted template that includes parameters and their corresponding values. This template can be used as the foundation for bulk updates

Keys and Values: The template includes the keys (parameter names) and their corresponding values, which the user has the ability to modify

Perform mapping with new parameters: Users can effectively modify the values of numerous parameters in the template and subsequently upload it to the interface. This interface will update the Excel-based economic model with the updated data

Logs for Mapping: The mapping log records both prior and new values and displays a log of all changes performed, improving transparency and traceability in model modifications

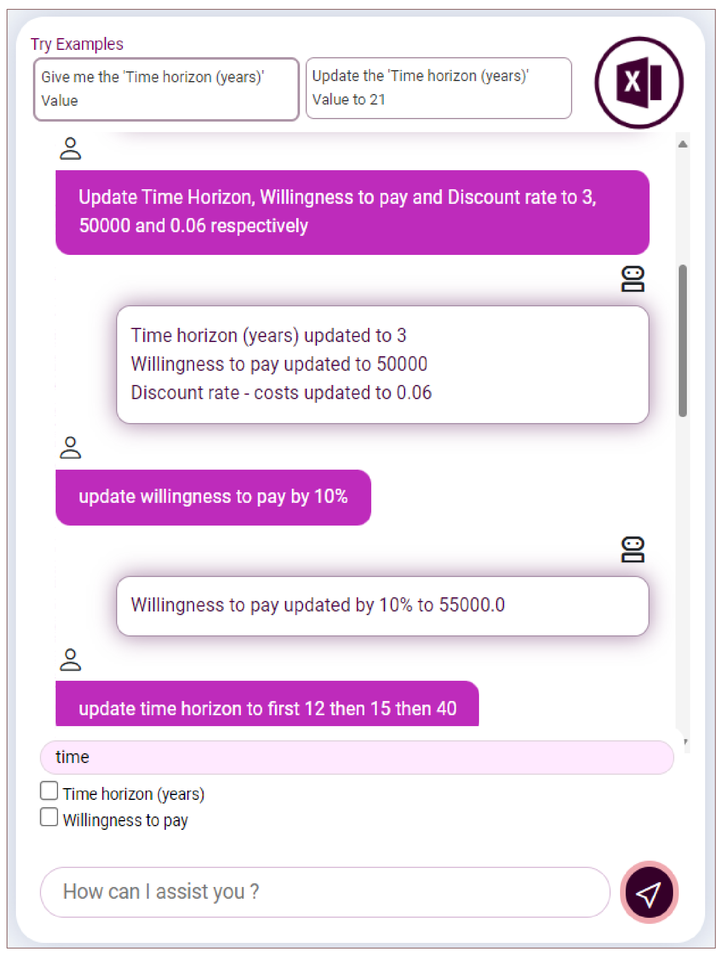
Table1. Test Cases (8 Samples)

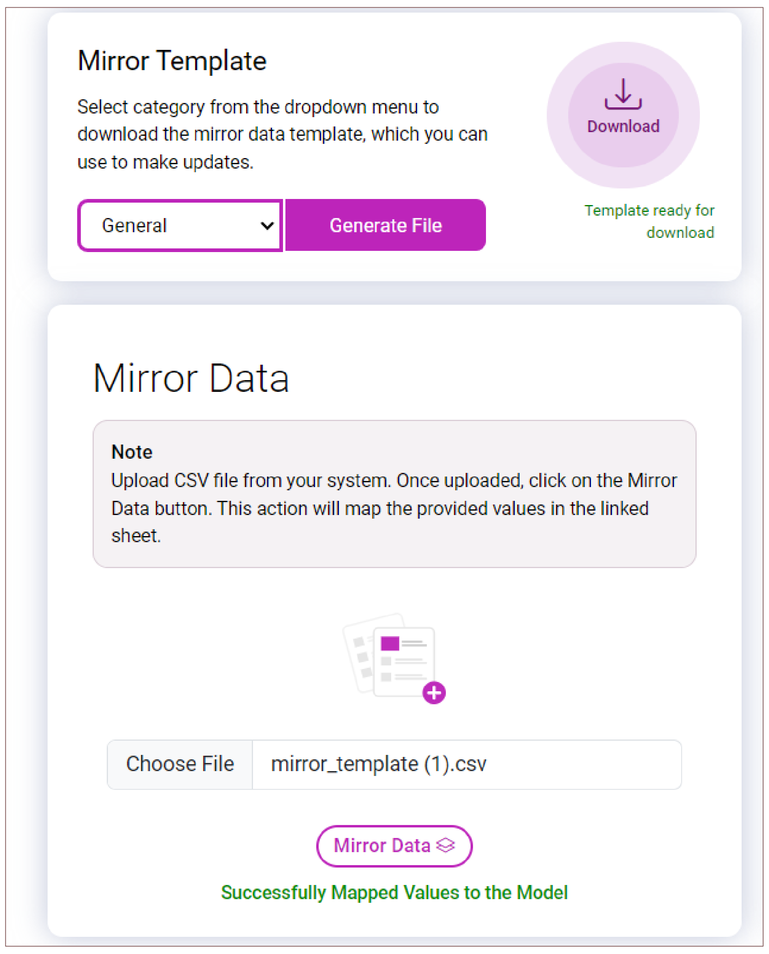
No.	Operation Type	Prompt Description	Complexity Level	Parameters Involved
1	Update	Update the Time Horizon to 5	Medium	Time Horizon
2	Retrieval	Show Willingness to pay	Simple	Willingness to pay
3	Update	Update Time Horizon, willingness to pay and Discount rate to 3, 5000 and 0.06 respectively	Complex	Time horizon, Willingness to pay and Discount rate
4	Update	Update willingness to pay by 10%	Medium	Willingness to pay
5	Update	Update Time Horizon to first 12 then 15	Complex	Time Horizon
6	Retrieval	Display willingness to pay and Time Horizon	Medium	Willingness to pay and time Horizon
7	Update	Update Discount rate to 2% if time Horizon is 40	Complex	Discount rate and Time Horizon
8	Retrieval	Show me the value of Discount rate, time Horizon and Willingness to pay	Medium	Time horizon, Willingness to pay and Discount rate

Results

- A comprehensive set of 30 prompts were designed to test the actual model updates: 10 for data retrieval and 20 for updates in the model. The AI interface correctly processed this pre-defined set of prompts: 10/10 for data retrieval and 20/20 for data updates in the Excel model
- The interface includes multi-parameter change functionality for country adaptations using an input sheet and has been tested with 20 distinct input sheets. The multi-parameter change functionality successfully updated the Excel model with the new values from the uploaded input sheets for all cases (Figure 5)
- This AI assistant's reliability was demonstrated by its consistent performance in all categories of calculations, including data retrieval operations (ranging from simple to multistep operations)

Figure 5. User interface





Mirroring Logs

Key	Label	Previous Value	New Value
timeHorizon	Time horizon (years)	21	10
wtp	Willingness to pay	30000	35000
discCosts	**Discount rate - costs**	**0.06**	**0.04**
discLYs	**Discount rate - LYs**	**0.035**	**0.02**

Conclusion

- This study introduces a novel method for integrating generative AI with Excel-based economic models. The integration of the LangChain framework and the Claude 3 Opus model (via AWS Bedrock) represents a substantial improvement in the application of AI in economic modelling.
- This study demonstrates that generative AI significantly increases efficiency by providing a much faster approach compared to manual adaptation of Excel-based economic models for country-specific needs. The Assistant UI offers a user-friendly interface, making Excel-based economic models more accessible to non-modellers.
- Future applications may leverage this interface as a unified platform for accessing various Excel-based economic models, ensuring a consistent and streamlined user experience.

Presented at ISPOR Europe 2024; November 17-20, 2024; Barcelona, Spain

Email: <shubhram.pandey@pharmacoevidence.com>

Copies of this poster are for personal use only and may not be reproduced without written permission of the authors.