ISPOR 2024
May 5-8, 2024

Automating economic modeling: the potential of generative AI for updating Excel-based cost-effectiveness models

Rawlinson, William MPhysPhil,[1] Teitsson, Siguroli MSc,[2] Reason, Tim MSc,[1] Malcolm, Bill MSc,[2] Gimblett, Andy Ph.D.,[1] Klijn, Sven L. MSc[3]

[1]Estima Scientific, London, United Kingdom; [2]Bristol Myers Squibb, Uxbridge, UK; [3]Bristol Myers Squibb, Princeton, NJ, USA

Agenda

- Introduction to large language models (LLMs) and their use in economic evaluation
- Case study: an application of large language models to Excel-based cost-effectiveness modelling

Large Language Models (LLMs)

LLMs are mathematical models that can generate human-like text in response to prompts

Write a one sentence slogan explaining the importance of HEOR

LLM

HEOR: Guiding Health Investments for Optimal Societal Benefit

LLMs have numerous applications in supporting, improving, and automating tasks currently performed manually in HEOR

LLMs and health economic modeling

At ISPOR Europe 2023, we presented on the capabilities of GPT-4 in automatically coding cost-effectiveness models in R

- We found that GPT-4 was able to accurately build replicas of two published health economic models in R [1]

However, most health economic models are currently built in Microsoft Excel

Therefore, an important step is to investigate applications of LLMs to Excel-based modeling tasks

[1] Reason, T., Rawlinson, W., Langham, J. et al. Artificial Intelligence to Automate Health Economic Modelling: A Case Study to Evaluate the Potential Application of Large Language Models. PharmacoEconomics Open 8, 191–203 (2024). https://doi.org/10.1007/s41669-024-00477-8

Excel vs R - from an LLM perspective

R models are made of text (code) which an LLM can interact with/interpret straight forwardly

Excel models are spreadsheets

	Cost of drug A	
	5	

What does the 5 represent?

Inferred through spatial relation to above cell

Applying LLMs in Excel modeling

We performed a case study to investigate an **application** for LLMs in Excel-based modeling

This case study focused on **cost-effectiveness model adaptation**

Model adaptation is the process of updating an existing health economic analysis to fit a new decision problem

This is frequently performed to generate country-specific cost-effectiveness analyses for HTA submissions

Global core model → Input adaptations and/or engine adaptations → Country-specific model

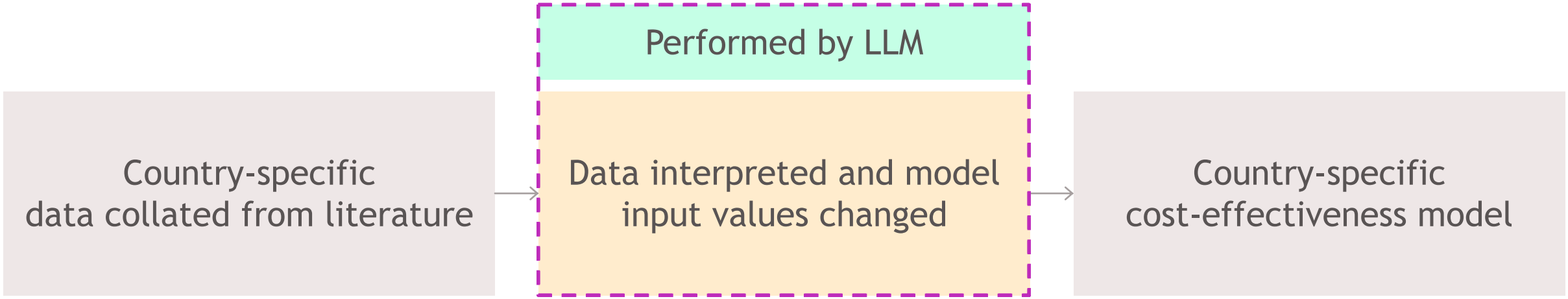
Study aim

Assess the capability of GPT-4* in automating adjustment of an HTA-ready Excel CEM from the setting of one country to another

*The gpt-4-1106-preview model (Nov 2023)

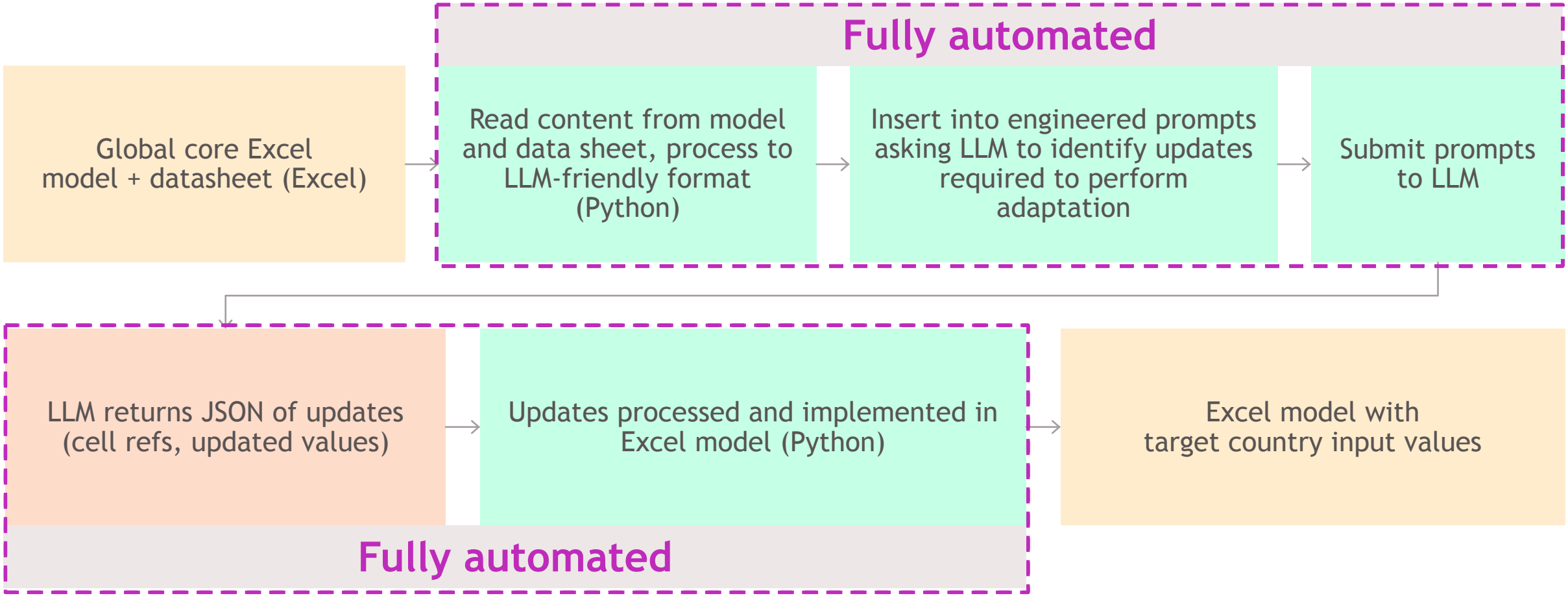
Scope of the case study

- Four our test case, we used GPT-4 to adapt a cost-utility model for muscle invasive urothelial carcinoma (MIUC) from a UK base case to a Czech Republic perspective
- The adaptation was restricted to **input values only** (no edits to the model engine were required)



Process map

We developed a fully automated process to adapt the inputs of a cost-effectiveness model from the setting of one country to another using a large language model



Closer look: the target country datasheet

Excel model +country-specific datasheet (Excel) → Read content from model and data sheet, process to LLM-friendly format (python) → Create prompts asking LLM to identify updates required to perform adaptation → Submit prompts to LLM → LLM returns list of updates (cell refs, updated values) → Updates processed and implemented in Excel model (python) → Excel model with country-specific input values

- We provided country-specific data to GPT-4 using an Excel sheet
- This used a mix of tables and natural language
- No cell references were provided (examples below, data provided in pink)

Natural language instruction	Czech republic resource costs provided	
Tabular data	Resource	Unit cost
	Urology consultant	475.2
	Urethroscopy	5637.6
	CT scan	1344.6
	Blood tests (kidney + PSA)	60.97
	Oncologist consult	452.52
	Nurse follow-up	475.2
	GP consultation	256.8
	Community nurse visit	163.08
	Health home visitor	163.08
	Dietician	0
	Palliative care doctor	256.8
	Palliative care nurse	163.08
	End of life costs	32726.1

Natural language instruction	Update discount rate for costs and QALYs to 3%			
Tabular data				

Closer look: the Global core Excel model

Excel model +country-specific datasheet (Excel) → Read content from model and data sheet, process to LLM-friendly format (python) → Create prompts asking LLM to identify updates required to perform adaptation → Submit prompts to LLM → LLM returns list of updates (cell refs, updated values) → Updates processed and implemented in Excel model (python) → Excel model with country-specific input values

Most CEMs have a central parameter sheet as switchboard between the user interface and the model engine (Typically 1,000+ rows of data)

We fed this sheet to GPT-4 rather than the entire model

GPT-4 used the sheet to determine the cell references for inputs, and their current values

Parameter	Value
Cohort size	1
Population	1
Perspective	Payer
Time horizon	30
% female	23.8%
Discount rate - costs	3.5%
Discount rate - QALY	3.5%
Discount rate - LY	0.0%
Mean age	65.60
Mean body weight - male	73.25
Mean body weight - female	73.25
Body surface area	1.79
Intervention	Nivolumab
Comparator 1	Observation
Comparator 2	Placeholder 1
Comparator 3	Placeholder 2
Comparator 4	Placeholder 3

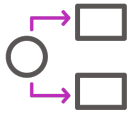
Closer look: Input file requirements

Excel model +country-specific datasheet (Excel) → **Read content from model and data sheet, process to LLM-friendly format (python)** → Create prompts asking LLM to identify updates required to perform adaptation → Submit prompts to LLM → LLM returns list of updates (cell refs, updated values) → Updates processed and implemented in Excel model (python) → Excel model with country-specific input values

Prior to running the fully automated process we:

Checked that the Excel model parameter sheet was descriptive

- E.g., there were a couple of instances where different parameters were labelled identically



Subdivided the model parameters and country-specific data sheet by category

- This was primarily to limit token usage (only 1/8th of model parameters were shown to GPT-4 at a time), but also improved performance

Closer look: Reading Excel data

Excel model +country-specific datasheet (Excel) → **Read content from model and data sheet, process to LLM-friendly format (python)** → Create prompts asking LLM to identify updates required to perform adaptation → Submit prompts to LLM → LLM returns list of updates (cell refs, updated values) → Updates processed and implemented in Excel model (python) → Excel model with country-specific input values

Python (**OpenPyXl**) was used to read and process data from the Excel model and data sheet

A **nested JSON string** proved a highly effective way to present the content of a worksheet, allowing LLMs to interpret large arrays of cells accurately


```
  },
  "Medical Resource Utilization: Unit Cost: Glucose Level Measurement": {
     "value coord": "G550",
     "value": 0.8389743297884098
  },
  "Medical Resource Utilization: Unit Cost: Glucose Tolerance Test (Oral)": {
     "value coord": "G551",
     "value": 0.9566045978538336
  },
  "Medical Resource Utilization: Unit Cost: Calcium Levels": {
     "value coord": "G552",
     "value": 1.0015461398825272
  },
  "Medical Resource Utilization: Unit Cost: Phosphate Levls": {
     "value coord": "G553",
     "value": 0.9909390155250861
  },
  "Medical Resource Utilization: Unit Cost: Thyroid Function Tests": {
     "value coord": "G554",
     "value": 2.6561354142814397
  },
  "Medical Resource Utilization: Unit Cost: Bone Densitometry (DEXA) Scan": {
     "value coord": "G555",
     "value": 87.39608697469555
  },
  "Medical Resource Utilization: Unit Cost: Liver Function Test": {
     "value coord": "G556",
     "value": 7.303157914943199
```

Closer look: Prompts and the LLM's output

"

- {Nested JSON representing country-specific data sheet}
- {Nested JSON representing Excel model central parameter sheet}
- What changes are required to adjust the Excel model to the country-specific setting?

"

Example output of GPT-4

['Update the average hourly wage to the new value provided', 'G889', '29.76']

Explanation for change	Cell ref	new cell value

Closer look: Automatically updating Excel files

Excel model +country-specific datasheet (Excel) → Read content from model and data sheet, process to LLM-friendly format (python) → Create prompts asking LLM to identify updates required to perform adaptation → Submit prompts to LLM → LLM returns list of updates (cell refs, updated values) → **Updates processed and implemented in Excel model (python)** → Excel model with country-specific input values

We used python (**xlwings**) to update input values in the Excel model based on the information returned by the LLM

It was straightforward to provide an audit of all changes made (for example, through highlighting changed cells) which allows for subsequent quality checks by health economists

A separate file to log changes could also be created

Case study results

The AI-generated adaptations were performed in 245 seconds

GPT-4 performed **62/64** required updates (two updates were missed)

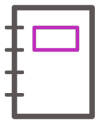
- Missed updates were due to an error in logic, where GPT-4 didn't change parameters that needed to be set to 0. This type of systematic error should be amenable through prompt engineering

No additional, unwanted updates were performed

Accuracy was **97%**

API costs were $0.64

Discussion



Our case study demonstrates the **technical feasibility** of using LLMs to automate editing of Excel-based models

This opens the door to many possible applications

Given that models are set up clearly, our study is promising early evidence that **accurate edits** of inputs can be achieved, which could **save significant time** given the size and complexity of Excel models

LLM-based adaptations could also be used to QC manually performed adaptations, **enhancing accuracy**

Limitations

We tested the performance of GPT-4 on a single adaptation

Accuracy, although high, did not reach 100%

Some set up (although minimal) was required. Greater set up may be required for models that are not well structured and labelled, although models can be designed a priori to be ‘AI-friendly’

Conclusions

To our knowledge, this is the first application of an LLM to make automated changes to an Excel-based health economic model

Future research could investigate the following improvements:

- Increasing accuracy to 100% through feedback-loops, and further prompt engineering

And the following potential applications:

- Adaptation/construction of model engines (e.g., building traces)
- Automated input collection
- Adaptation of technical reports for Excel models (see poster EE205)
- Using LLMs to QC Excel-based models