Assessing the Generalizability of Automating Adaptation of Excel-based Cost-Effectiveness Models Using Generative AI

[HTA156]

William Rawlinson,[1] Siguroli Teitsson,[2] Tim Reason,[1] Bill Malcolm,[2] Andy Gimblett,[1] Sven Klijn,[3]

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

Objectives

Adapting cost-effectiveness models

- Model adaptation is the process of updating an existing health economic analysis to fit a new decision problem.
- Model adaptation is most frequently used to adapt a 'core' health economic analysis to the setting of different countries for use in Health Technology Assessments (HTAs).
- Adaptations vary in scope, but can involve updating parameter values (e.g., updating resource costs to reflect the new country), updating the model engine (e.g., adding additional options for survival modelling), and adding or removing comparators.

Automating model adaptation

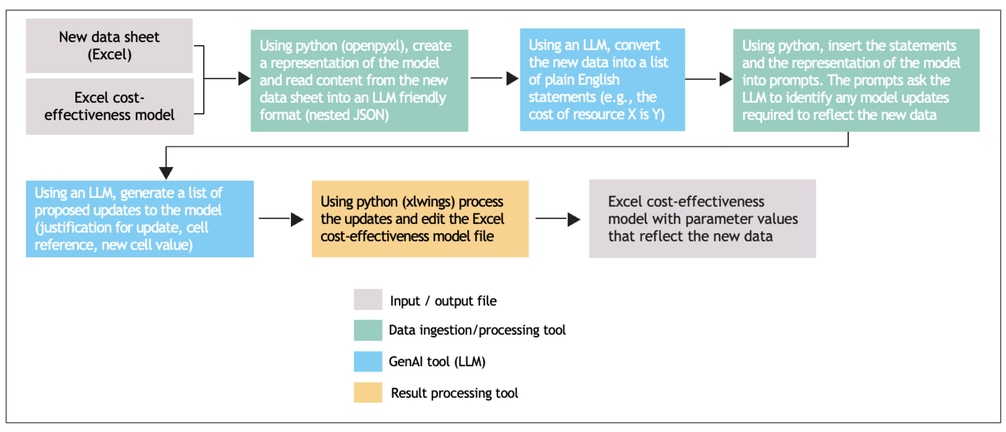
- LLMs have exhibited potential in automation of health economic modelling tasks [1]
- A previous study [2] described a method 'LLMAdapt' that uses a large language model (LLM) to automatically adapt the parameter values of an Excel-based cost-effectiveness model (CEM) from the setting of one country to another. The study found a high level of accuracy (97%) for one test case which used an HTA-ready cost-effectiveness model.
- The original study evaluated the performance of LLMAdapt on a single model. It is important to test the generalisability of AI-based solutions to ensure they are not overly specific and can work across a range of examples.
- The objective of this study was to assess the generalisability of LLMAdapt across two distinct disease areas and countries.

Methods - LLMAdapt

LLMAdapt

- LLMAdapt is a GenAI toolchain (a sequence of software tools designed to automate a complex task) which aims to automatically update parameter values in an Excel-based cost-effectiveness model to reflect new data. The toolchain was originally developed for ISPOR US 2024 [2] and was given several updates for this study, including adding self-consistency prompting and enhancing the task decomposition approach.
- The toolchain uses two input files, the Excel cost-effectiveness model and a new datasheet (also in Excel format). The output, an updated Excel cost-effectiveness model, is generated without human intervention.
- The process is powered by an LLM, which interprets the new data sheet and the Excel model, and works out what parameter values in the Excel model should be updated to reflect the new data.
- A top-level diagram of the improved LLMAdapt toolchain is shown in **Figure 1.**

Figure 1. LLMAdapt process map (fully automated)

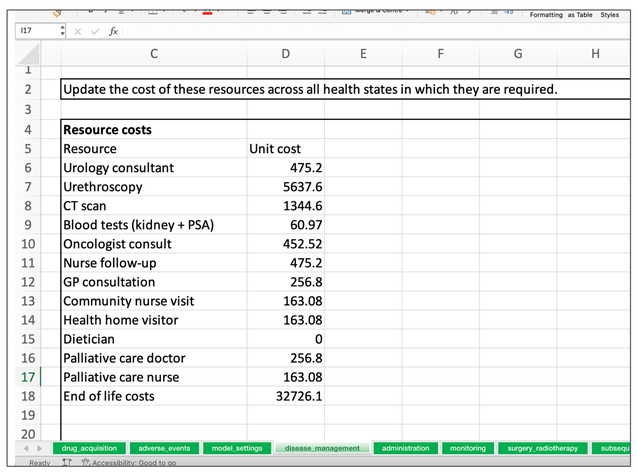


Input files

- The toolchain uses two input files.

1) The first input file, the new data sheet, is built in Excel. The new data sheet uses a mixture of tables and natural language, and aims to mimic the format in which data for model adaptation might be collected during a targeted literature review (see **Figure 2**). Some formatting restrictions are imposed due to the challenges in AI-interpretation of inconsistently structured spreadsheet data. Importantly, the new data sheet doesn't contain any information about the cost-effectiveness mode. The LLM has to work out how the model should be edited to reflect the new data.
2) The second input file is the Excel cost-effectiveness model itself. Most CEMs have a central parameter sheet to route raw parameter data through sensitivity analyses. This sheet is fed to the LLM rather than the entire model, as it provides efficient access to the model's current parameter values and the cell reference locations in which they can be edited (via backtracking formulae). The process works best with a well-labelled parameter sheet.

Figure 2. New data sheet



Update the cost of these resources across all health states in which they are required.

Resource costs	
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

Reading Excel data

- Python (OpenPyxl) was used to read data from the Excel input files.
- The processing tools used nested JSON strings to communicate structured spreadsheet data, which we found facilitates accurate interpretation of large arrays of cells [ref] (see **Figure 3**).

Figure 3. Nested JSON Excel data


```
},
"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
```


Prompting

- For this study, we used GPT-4-1106-preview (a state-of-the-art LLM from OpenAI's GPT series) to power the toolchain.
- The prompting strategy was based on the principle of task decomposition. An LLM will usually perform a complex task more accurately when tackling each component of the task in isolation, rather than in response to a single prompt.
- Two seperate GenAI tools were used.

1) The first tool uses an LLM to convert tabular data from the new data sheet into plain English statements.
2) The second tool shows an LLM the nested JSON representation of the CEM central parameter sheet, and the plain English statements describing the new data, and asks the LLM to generate a list of proposed updates to the CEM with justification (see **Figure 4**).

- To further decompose tasks into isolated components, data from the new data sheet were presented to the LLM one row at a time.

Methods - LLMAdapt

Figure 4. Core prompt

Structure of core prompt

{Plain English statement representing row of new data sheet}

{Nested JSON representing CEM's central parameter sheet}

What parameter values should be changed to adjust the model to reflect the new data? Return your output in the following format:

[Justification for your change, Cell reference requiring change, new cell value]

Structure of response

['Update GP visit cost to reflect the provided UK data', 'C58', '£56']

Justification for change | Cell ref | New cell value

Self-consistency

- In addition to the prompting strategy, we implemented a self-consistency approach [3] to maximise the accuracy of the toolchain by reducing non-systematic error.
- Each prompt was submitted to the second GenAI tool five times, and only responses that were observed in at least 80% of cases were taken forward (see **Figure 5**)

Figure 5. Self-consistency

Submit the same prompt 5 times and record the responses

['Update GP visit cost to reflect the provided Czech Republic data', 'C58', '£56']

['Change value for GP cost, as it does not match the Czech Republic data', 'C58', '£56']

['Update the GP visit cost based on the new Czech Republic data', 'C70', '£56']

['Edit the cost of a GP visit to reflect the new Czech Republic data', 'C58', '£56']

['Update GP visit cost to reflect Czech Republic data', 'C58', '£56']

Take forward only responses that occured at least 80% of the time

['Update GP visit cost to reflect the provided Czech Republic data', 'C58', '£56']

Updating Excel files

- To conclude the toolchain, python (xlwings) was used to update the Excel file based on the output of the second GenAI toolchain.
- Updates were automatically performed in the location at which the raw parameter value was defined and any updated parameters were highlighted in the central parameter sheet of the CEM to aid subsequent human review.

Methods - Case study design

- LLMAdapt was tested on two distinct HTA-ready models covering two distinct disease areas (one in muscle-invasive urothelial carcinoma (MIUC) and one in myelodysplastic syndrome (MDS).
- The MIUC model was adapted from a UK base case to a Czech Republic perspective, and the MDS model from a UK base case to a US Medicare perspective.
- Categories of country-specific data used in the adaptations included: therapy costs, discount rates, patient characteristics, medical resource costs, medical resource consumption, health state utilities, disutilities, and subsequent therapy utilisation and cost.
- To evaluate the performance of LLM adapt, a human health economist created a list of required parameter updates based on the new data sheet, and these were compared against the updates made by the automated toolchain.

Results

- The human health economist identified that 102 and 199 parameter values should be updated based on the new data sheet for the MIUC and MDS models, respectively.
- Without human intervention LLM adapt identified and successfully performed 101/102 (99.0%) and 198/199 (99.4%) of those updates.
- The two errors were ommissions, where the LLM failed to identify a parameter that should have been updated.
- The execution time for the toolchain was 132 seconds for the MIUC model and 207 for the MDS model.
- Compute costs were $9.44 and $8.54, respectively.

Table 4. Result table

Model	Parameters updated	Accuracy score	Execution time (seconds)	Compute cost
MIUC	101/102	99.0%	132	$9.44
MDS	198/199	99.4%	207	$8.54

Discussion

- LLMAdapt maintained high accuracy (99.0% and 99.4%) across hundreds of parameter value updates to two HTA-ready models in two distinct disease areas, providing evidence for the generalisability of LLM-based parameter adaptations of Excel cost-effectiveness models.
- Accuracy results improved relative to the first version of LLMAdapt (presented at ISPOR US 2024). The improvement derived from the use of a self-consistency approach and re-design of the toolchain structure, illustrating the importance of method design when utilising generative AI for HEOR-related applications.
- The improved process led to increased compute costs (API fees). However, the costs were still very low (<$10) and execution time was not impacted. Compute costs can be reduced by dividing the central parameter sheet into granular categories (e.g., drug acquisition costs, monitoring frequencies, utilities) so that the LLM doesn't need to view the entire sheet when calculating each parameter adaptation. This explains why the MIUC model adaptations were more costly than the MDS adaptations, despite the fact that half the number of parameters were updated.
- A single parameter value update was missed for each adapted cost-effectiveness model. The errors were observed to occur where several updates were required to reflect a single data point. A potential mitigation strategy could be to include a reflection step, where the core LLM (or another LLM) assesses whether the initial proposed parameter updates are comprehensive.
- The process of adapting cost-effectiveness models to reflect new data is time consuming and error prone when performed manually. In addition to country adaptations, similar processes are performed for updated database locks.
- Methods such as LLMAdapt have the potential to reduce this manual burden, enabling adapted models to be produced at a lower cost and in more rapid timeframes.
- Some structural restrictions were placed on the input files (new data sheet, Excel CEM) to maximise the capabilities of the LLM in interpreting the Excel data.
- Further research could explore 'LLM-friendly' Excel model layouts. This could be the key to unlocking the full potential of LLMs in working with Excel models (for example, enabling LLMs to edit Excel model engines or review model formulae). An ideal format would not only be LLM-friendly, but also easy for a human to work with and develop.

Key takeaways

- LLMAdapt is a method that uses a large language model (LLM) to automatically adapt the parameter values of an Excel-based cost-effectiveness model (CEM) to reflect new data.
- LLMAdapt was provided country-specific data in Excel format and automatically adapted two HTA-ready models without human intervention.
- LLMAdapt identified and successfully performed 101/102 (99.0%) and 198/199 (99.4%) of the required updates. Execution time was 132 and 207 seconds, respectively, and compute costs were $9.44 and $8.54.
- This case study provides evidence of the generalisability of this method across different disease areas.
- Further research could investigate 'LLM-friendly' Excel model layouts to enable more advanced modelling tasks (e.g., reviewing the formulae in a model engine)

References

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
2. Rawlinson W, Teitsson S, Reason T, Malcolm B, Gimblett A, Klijn SL. Automating Economic Modelling: Potential of Generative AI for Updating Excel-Based Cost-Effectiveness Models. Value in Health. 2024;27(6)
3. Wang et al. Self-Consistency Improves Chain of Thought Reasoning in Language Models. ArXiv, 21 Mar 2022. https://doi.org/10.48550/arXiv.2203.11171

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

Email: will.rawlinson@estima-sci.com