ScienceDirect

Contents lists available at sciencedirect.com
Journal homepage: www.elsevier.com/locate/jval

Themed Section: Artificial Intelligence in Health Economics and Outcomes Research

Generative Artificial Intelligence to Automate the Adaptation of Excel Health Economic Models and Word Technical Reports

William Rawlinson, MPhysPhil, Siguroli Teitsson, MSc, Tim Reason, MSc, Bill Malcolm, MSc, Andy Gimblett, PhD, Sven L. Klijn, MSc

ABSTRACT

Objectives: In health economics and outcomes research (HEOR), many repetitive tasks could be performed by large language models (LLMs), including adapting Excel-based health economic models and associated Word technical reports to a new setting. However, it is vital to develop robust methods so that the LLM delivers at least human-level accuracy.

Methods: We developed LLM-based pipelines to automate parameter value adaptations for Excel-based models and subsequent reporting of the model results. Chain-of-thought prompting, ensemble shuffling, and task decomposition were used to enhance the accuracy of the LLM-generated content. We tested the pipelines by adapting 3 Excel-based models (2 cost-effectiveness models [CEMs] and 1 budget impact model [BIM]) and their associated technical reports. The quality of reporting was evaluated by 2 expert health economists.

Results: The accuracy of parameter value adaptations was 100% (147 of 147), 100% (207 of 207), and 98.7% (158 of 160) for the 2 CEMs and 1 budget impact model, respectively. The parameter value adaptations were performed without human intervention in 195 seconds, 245 seconds, and 189 seconds. For parameter value adaptations, the application programming interface costs associated with running the pipeline were $13.36, $6.48, and $2.65. The accuracy of report adaptations was 94.4% (17 of 18), 100% (54 of 54), and 95.1% (39 of 41), respectively. The report adaptations were performed in 128 seconds, 336 seconds, and 286 seconds. For report adaptations, the application programming interface costs associated with running the pipeline were $1.53, $4.24, and $4.05.

Conclusions: LLM-based toolchains have the potential to accurately and rapidly perform routine adaptations of Excel-based CEMs and technical reports at a low cost. This could expedite health technology assessments and improve patient access to new treatments.

Keywords: artificial intelligence, large language models.

VALUE HEALTH. 2025; 28(11):1683–1689

Highlights

- Many countries require a health technology assessment of new health interventions, which often requires adapting a "global" economic model and technical report from the setting of the reference country to the setting of the new country.
- We developed robust methods to automatically adapt global Excel-based health economic models and associated Word technical reports using large language models. Using these methods, routine adaptations of Excel-based models and technical reports were performed accurately and rapidly at a low cost.
- Large language models have huge potential for automating tasks that are currently performed manually in health economics and outcomes research, which could greatly expedite the health technology assessment process and improve timely patient access.

Introduction

Providing patients with timely access to new treatments is vital for improving health outcomes.[1] Many countries require a health technology assessment (HTA) of new health interventions, which requires pharmaceutical companies to evaluate the cost-effectiveness of new interventions against already available treatments.[2] This is a time-consuming and resource-intensive process that typically involves, among other requirements, developing an Excel-based economic model and providing technical documentation of the model's methods, input data, and results. To help minimize the time and resources required to generate economic evaluations across multiple countries, a "global" economic model and technical report may be developed, most often from the perspective of 1 country's healthcare system and subsequently adapted to the setting of other countries.[3]

"Model adaptations" vary in complexity and scope, but typically involve updating parameter values (eg, treatment costs), updating the model engine (eg, adding age-adjusted utilities), and adding or removing comparator treatments.[3] Any updates made to the economic model and associated changes in the model results must be reflected in the text, tables, and figures of the adapted technical report. Therefore, generating country-specific economic evaluations by adapting global economic models and technical reports can still be a costly and time-consuming process.

Large language models (LLMs) are large-scale, pretrained, statistical language models based on neural networks.[4] LLMs use learned statistical relationships to predict the next "token" in a sequence, producing human-like responses to prompts. LLMs could enable automation of tasks that are currently performed manually in health economics and outcomes research (HEOR),[5-7] which could greatly enhance efficiency and reduce costs.[8-12] For example, LLMs have been previously applied to automate the programming of health economic models in R (language and environment for statistical computing. R Foundation for Statistical Computing, Vienna. Austria), based on natural language specifications.[12]

The aim of this paper was to assess the feasibility of automatically adapting global Excel-based models and Word-based technical reports from the reference market to another country using LLMs. We focused on 2 adaptation tasks that are routinely performed: adapting parameter values in the economic model and adapting the reporting and discussion of model results in the technical report. We developed an LLM pipeline to automate each task. We assessed the performance of the pipelines by adapting 3 HTA-ready Excel models (2 cost-effectiveness models [CEMs] and 1 budget impact model [BIM]) and their associated technical reports. This article builds on previous research presented at conferences, using a greater number of test cases, improved pipelines, and reporting methods in greater detail.[13-15]

Methods

Development

An LLM pipeline is a piece of software that uses interactions with LLMs, as well as traditional programming, to automate a task. We developed 2 LLM pipelines, 1 for adapting parameter values in an economic model and 1 for adapting the reporting and discussion of model results in a technical report. We built the pipelines in Python (version 3.12.7; Python Software Foundation) which allows interaction with LLMs via application programming interface (API) calls to providers such as OpenAI and Anthropic (which can be hosted in a secure environment such as Microsoft Azure or Amazon Bedrock). As such, no dedicated computing resources are required to run them. With that said, running the pipelines incurs an API cost (based on rates set by providers); the API costs for each test case are presented in the results. During development, the pipelines were tested on a mix of dummy data and some excerpts from the first test case. Because this presents a danger of overfitting the pipelines, we chose 3 varied test cases to ensure an adequate level of generalizability.

Model Adaptation Pipeline

The model adaptation pipeline is designed to automate the following task: adapting parameter values in an Excel health economic model. When parameter values are adapted manually, the following process is used. First, country-specific data are collected, for example, through literature review, consulting local databases, and/or interviewing local clinical experts. Second, the country-specific data identified are then provided to a health economist. Third, the health economist leverages their understanding of the model to identify parameters that should be updated to reflect the new data and to locate the cells in which the parameters can be updated. This includes performing any required calculations on the country-specific data, such as converting units to match a parameter's definition.

The model adaptation pipeline aims to automate the role of the health economist in the abovementioned process, starting from the point at which country-specific data are made available. The pipeline works in 3 steps:

1. Understand the Excel model and the country-specific data.
2. Locate cells in the Excel model that should be updated to reflect the new data (performing calculations, if necessary).
3. Update the values of those identified cells.

The model adaptation pipeline requires a global Excel model and an Excel file containing country-specific data as inputs. Broadly, the pipeline first uses the *openpyxl* Python package to extract information from the Excel model (parameter names, parameter cell locations, current parameter values) and from the country-specific data file (country-specific data points). Then, an LLM is used to interpret the country-specific data, providing plain English descriptions. In step 3, an LLM is used to identify parameters in the global CEM that should be updated to reflect the country-specific data, locate their cells in the Excel model, and suggest new values for those cells. Finally, the xlwings Python package is used to process the LLM's responses from step 3 into actual changes in the Excel model file. The structure of the model adaptation pipeline and a detailed example workflow for mock data are shown in Figure 1. Further key information pertaining to the design and functionality of the pipeline is provided below.

Input File Requirements

The model adaptation pipeline we designed needs the 2 input files (Excel country-specific data file and Excel global model) to comply with a set of specific requirements to function effectively. This was necessary given the significant heterogeneity of modeling and data collection approaches that are used in HEOR. Full details on requirements are provided in the Appendix 1 in Supplemental Materials found at https://doi.org/10.1016/j.jval.2025.05.020, and the country-specific data files used in each of the test cases are provided as additional materials. However, the 2 key requirements are presented below.

First, the global model must use a "central parameter sheet," which is a sheet (such as a sensitivity analysis filter) through which all model parameters are routed and labeled and which uses a simple structure (1 row per parameter name/value). Although this approach is very common, some Excel models may require manual setup to meet this requirement.

Second, for optimal performance, the central parameter sheet should present parameters in categories (eg, "drug cost parameters," "adverse event rate parameters"). The country-specific data file can then be organized into separate sections, each containing data relevant to a specific category. This means that, in step 3 of the process, the LLM can be shown only parameters from the relevant category, rather than all parameters in the model, which reduces prompt length dramatically and therefore increases performance. This approach was used for the test cases later in the article.

LLM

The claude-3-5-sonnet-20241022 model was used in all LLM-based processes in the model adaptation pipeline. The following 2 sections ("Task Decomposition" and "Prompting") describe key methods that were used to maximize the accuracy and reliability of these LLM-based processes.

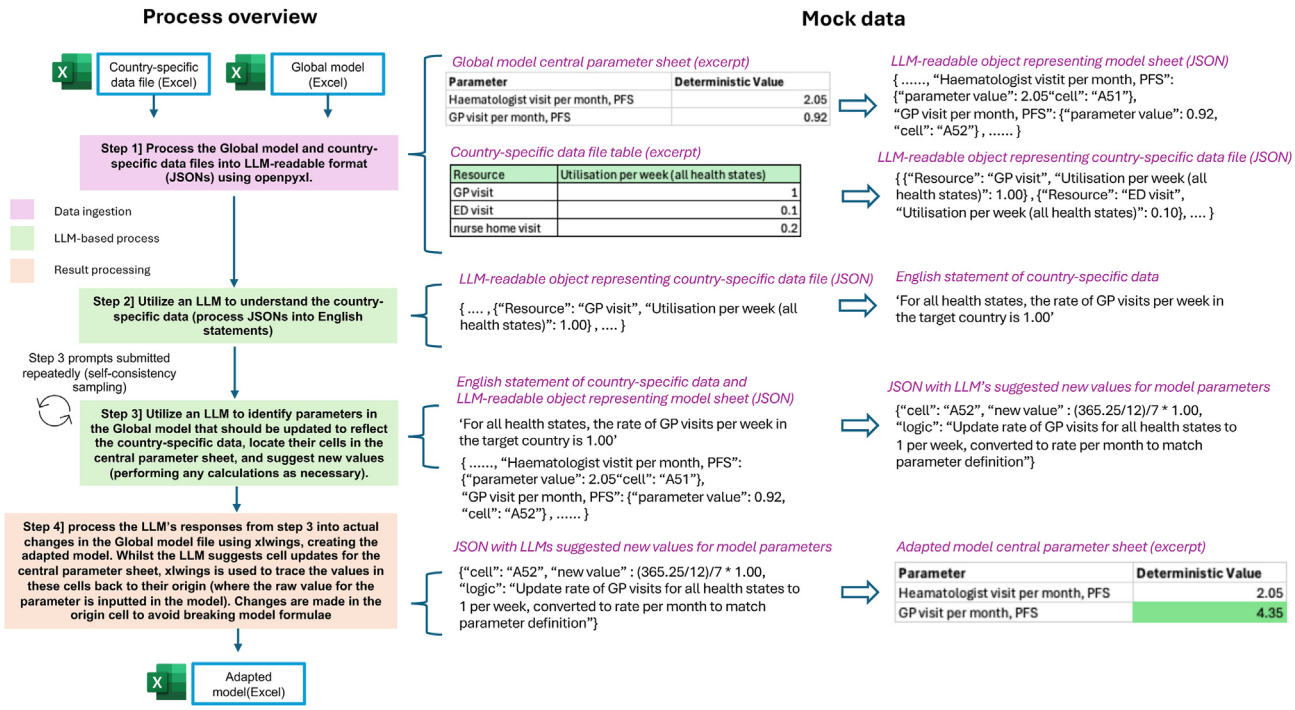
Task Decomposition

Task decomposition means splitting tasks into their constituent parts and can improve the ability of LLMs to complete lengthy, complex tasks.[16] Task decomposition was implemented in several places in the model adaptation pipeline. When understanding the country-specific data in step 2 (Fig. 1), the LLM was prompted with only a single row of data at a time. When suggesting model updates in step 3, the LLM was prompted with only a single statement about a single country-specific data point at a time. Furthermore, the rationale for including step 2 was so that the LLM did not have to both interpret the output of step 1 and suggest the resulting model update in a single interaction.

Prompting

Two techniques were used to enhance the prompting for the model adaptation pipeline. Both were applied at step 3 (Fig. 1). This step required special attention as the LLM had to accurately

Figure 1. Model adaptation toolchain diagram.



ED indicates emergency department; GP, general practitioner; LLM, large language model; PFS, progression-free survival.

extract information from a very large JSON containing model parameter information (eg, this could contain more than 1000 parameters), and LLMs can produce lower-quality responses as the number of tokens in a prompt increases.[17] Furthermore, in step 3, the LLM looks for parameters to update from an ordered list (the JSON) a task that shares similarities with a multiple-choice problem. LLMs have been observed to show option order bias in multiple-choice problems, favoring earlier options, and we observed a similar tendency for the LLM to omit updates for parameters in the later third of the JSON in initial testing.[18]

Figure 2. Step 3 prompt structure—model adaptation pipeline.

Demarked by [Parameter sheet] is a dictionary that lists parameters in an Excel model.

Demarked by [Country-specific data] are some new data for the Excel model. Your task is to provide a list of updates for the Excel model.

…. *Further specific instructions* ….

The updates should be returned in a specific format, see the below example:

[{"logic": 'logic for update 1', "cell": 'C567', "new value": '21'},{"logic": 'logic for update 2', "cell": 'C23', "new value": '33'}]

…. *Further specific instructions* ….

[Parameter sheet]
>JSON representing Global CEM central parameter sheet<

[Country-specific data]
>Statement describing country specific data point<

Now, answer the question. Let's think in steps to ensure we don't miss any updates. Start your answer by stating 'first we will look through the entire parameter sheet to ensure we don't miss any of the required updates'.

The first technique to address these challenges was a basic implementation of chain-of-thought prompting, with the following text included to elicit a step-by-step response from the LLM: "Let's think in steps to ensure that we don't miss any updates. Start your answer by stating 'first we will look through the entire parameter sheet to ensure we don't miss any of the required updates'" (Fig. 2).

The second technique we used at step 3 was "shuffled" self-consistency prompting. Self-consistency prompting means submitting the same prompt multiple times and taking forward only the most common answer[19] and can help reduce nonsystematic error. For each iteration, we also "shuffled" the prompt, changing the order in which the JSON presented the model parameters. This helped to mitigate the impact of option order bias on the LLM's most common answer.

Report Adaptation Pipeline

The report adaptation pipeline is designed to automate the following task: adapting the reporting and discussion of model results in the technical report to reflect the adapted Excel model results. When technical reports are adapted manually, a health economist will typically copy result tables and figures from the adapted Excel model into relevant sections of the technical report and update free text to reflect the new set of results.

The report adaptation pipeline aims to automate this process. The pipeline works in 3 core steps:

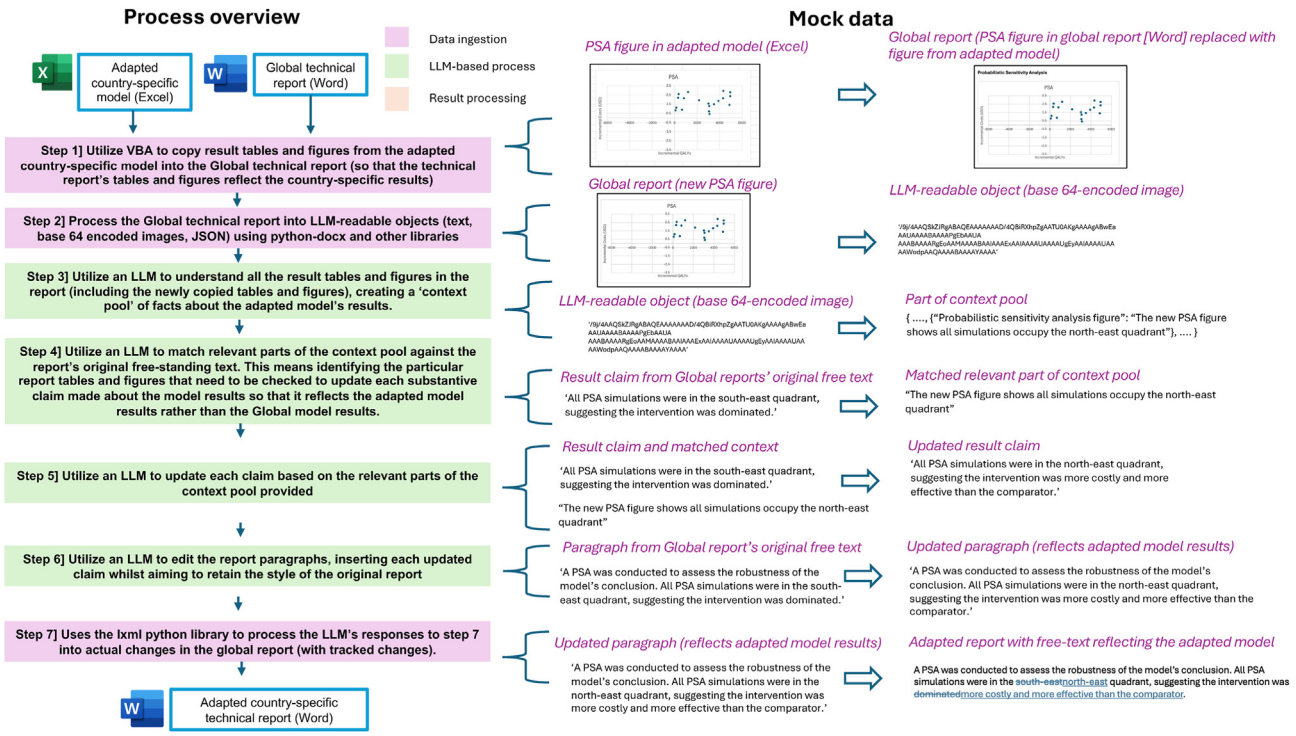
1. Copy new result tables and figures from the adapted Excel model into the technical report.
2. Interpret the new result tables and figures, creating a repository of information about the adapted model results (a "context pool").
3. Using the context pool, edit free text in the report to ensure that claims made about the model results reflect the adapted model results, rather than the original model results.

The report adaptation pipeline requires an adapted Excel model and Word global technical report as inputs. Broadly, the pipeline first uses Visual Basic for Applications to copy result tables and figures from the adapted Excel model into the global technical report. Then, tables, figures, and free text from the technical report are processed into LLM-readable objects (JSONs for tables, base 64-encoded images for figures, and text for free text) using Python. An LLM is then used to interpret the report's result tables and figures (including the newly copied tables and figures) to create a context pool describing the adapted model results. Then, we use an LLM to match parts of the context pool against the report's free text, essentially identifying the particular tables and figures that need to be checked to update each claim made about the model results. With this matched context, an LLM then updates the report's free text, ensuring that all claims about model results reflect the adapted model results, while aiming to retain the style of the original report. Finally, the lxml Python library is used to insert the LLM's updated text into the technical report (with tracked changes) creating the adapted report. The structure of the model adaptation pipeline and a detailed example workflow for mock data are shown in Figure 3. Technical details on specifics such as handling cross-references are provided in the Appendix 1 in Supplemental Materials found at https://doi.org/10.1016/j.jval.2025.05.020. Further key information pertaining to the design and functionality of the pipeline is provided below.

Input File Requirements

As for the model adaptation pipeline, the report adaptation pipeline needs the 2 input files (Excel-adapted model file and Word

Figure 3. Report adaptation toolchain diagram.



LLM indicates large language model; PSA, probabilistic sensitivity analysis; VBA, Visual Basic for Applications.

global report) to comply with a set of specific requirements to function effectively, to handle the significant heterogeneity of modeling and reporting approaches that are used in HEOR. As for model adaptation, full details on requirements are provided in the Appendix 1 in Supplemental Materials found at https://doi.org/10.1016/j.jval.2025.05.020, and the key requirements are presented below. Please note that the artificial intelligence (AI)-adapted reports from the 3 test cases are provided as additional materials.

The central requirement is needed for step 1, where result tables and figures are copied from the adapted Excel model to the global report. To facilitate this process, the Excel model should contain "live" result tables and figures that are suitable for display in the report. These elements need to be linked to model result cells, so they always display the current model results. The live tables and figures must also be identified using named ranges that link to corresponding tables and figures in the global technical report (identified using bookmarks, these are the tables and figures that will be copied over).

LLM

The report adaptation pipeline used 3 different LLMs. Claude-3-5-sonnet-20241022 was used for steps 4 and 6, gpt-4o-2024-08-06 was used for step 3, and o1-preview-2024-09-12 was used for step 6. Gpt-4o-2024-08-06 and o1-preview-2024-09-12 were brought in for specific purposes. GPT-4o has particularly impressive long context performance[20] and performed well in extracting all details when interpreting figures and tables in step 4. Step 6 may require a long paragraph to be edited with many different updated claims, in a single interaction. As a reasoning model, o1-preview proved well suited to staying on track during this lengthy, multicomponent task.

Task Decomposition

Task decomposition was implemented in several places in the report adaptation pipeline. First, elements from the global report (paragraphs, figures, tables) were interpreted one at a time in step 3. Second, steps 4 and 5 were performed separately for each result claim identified in the original report's free text. For example, the sentence "The highest total cost was drug acquisition costs, which were $5000 for treatment A, and $3000 for treatment B." would be decomposed into 3 separate result claims.

Prompting

No specific techniques (aside from task decomposition) were used to enhance the prompting for the report adaptation pipeline. However, it is important to note that for step 5, we prompted claude-3-5-sonnet with instructions to state when it could not conclusively verify a result claim based on the context pool. This might occur if the report describes a result that is not included in the report tables or figures. We asked the model to include the marker "[unable to verify]" in this case, to facilitate rapid subsequent human review of the adapted report.

Case Studies

We assessed the performance of the pipelines by adapting 3 Excel-based HTA-ready models and their associated technical reports. The models represented different disease areas and 3 different model types: a CEM with a Markov structure, a CEM with a partitioned survival structure, and a BIM. Because of data sensitivity, it is not possible to provide the models, country-specific data files, and technical reports in full in this article. However, redacted excerpts from each of these files are provided in the Supplemental Materials found at https://doi.org/10.1016/j.jval.2025.05.020. This includes the full result and discussion section of each adapted report, each of the country-specific data files used to adapt the models, and excerpts from each of the models.

To evaluate the performance of the model adaptation pipeline, we ran the pipeline on each of the 3 Excel models and compared the output of the pipeline with manual adaptations performed by a human health economist with 4 years of experience (a coauthor of the article). We used country-specific data files with randomized data that covered parameters that would typically be updated during an adaptation (costs, healthcare resource use, utilities, adverse event rates, etc).

To evaluate the performance of the report adaptation pipeline, we ran the pipeline on the AI-adapted models and their associated global technical reports. The free text in each AI-adapted report was assessed by 2 health economists (coauthors of the article with 10+ years' experience). Note that tables and figures copied across from the adapted model were not assessed, because this process uses deterministic programming rather than relying on LLMs. The report text was evaluated in 2 domains: factual accuracy and tone. Accuracy was measured by identifying all substantive statements about model results and assessing whether these statements were correct in the final AI-generated reports, with scores aggregated across the 2 assessors. Tone was assessed qualitatively, based on whether the AI-generated text used appropriate phrasing and retained the word choice of the original text where possible.

Results

Model Adaptations

The parameter value adaptations were performed without human intervention in 195 seconds, 245 seconds, and 189 seconds. API costs were $13.36, $6.48, and $2.65. API costs were higher for the first model because it used wider categories of parameters. The accuracy of parameter value adaptations was 100% (147 of 147), 100% (207 of 207), and 98.7% (158 of 160) for the 2 CEMs and 1 BIM, respectively. The 2 errors were traceable to the logic provided as part of the proposed updates. The errors involved misinterpretation of the scope of a data point, because of an incorrect assumption about which comparators could be classified as chemotherapy. Interestingly, the errors both appeared in only one of the self-consistency responses (step 3). This suggests that techniques to resolve or flag "conflicts" across the multiple responses could reduce the rate and/or impact of errors (see Discussion).

Report Adaptations

The reporting adaptations were performed in 128 seconds, 336 seconds, and 286 seconds. API costs were $1.53, $4.24, and $4.05. The factual accuracy of the report adaptations was 94.4% (17 of 18: 6 statements correctly edited, 11 statements correctly left unedited, 1 statement incorrectly edited), 100% (54 of 54: 46 statements correctly edited, 8 statements correctly left unedited, 0 statements incorrectly edited), and 95.1% (39 of 41: 29 statements correctly edited, 10 statements correctly left unedited, 2 statements incorrectly edited), respectively. The error in the first report stemmed from the LLM failing to identify a substantive claim about model results. Both errors in the third report were caused by a misinterpretation of common terms in the BIM: first, what is meant by "total budget" and second what is meant by "the world with/without" the intervention. This suggests the toolchain might benefit from a dynamic repository of HEOR definitions (see Discussion). The LLM marked that it was unable to verify 1 statement

in the first report, 9 statements in the second report, and 3 statements in the third report, because of a lack of context. Please note that the AI-adapted reports from the 3 test cases are provided as additional files in the Supplemental Materials found at https://doi.org/10.1016/j.jval.2025.05.020.

For most adaptations, the tone and word choice of the edits were very closely aligned to the original paragraphs. For example, in report 2, the following text: "Breakdown of QALYs by health state further indicate that the vast majority of QALYs are generated in the health state 2 health state, 80% in the Treatment A arm and 71% in the Treatment B arm" was updated to "The breakdown of QALYs by health state further shows that the majority of QALYs are generated in the health state 1 health state, with 77.0% in the Treatment A arm and 65.9% in the Treatment B arm." by the toolchain (please note that black highlighting indicates redaction applied after the AI-generated report was created). Tone was also maintained for most adaptations where new interpretations were required. For example, the following text in report 3: "These savings are offset by the higher drug acquisition and administration costs for the scenario with Treatment A and higher costs for AEs. Total budget impact is estimated at 17.11 million" was updated to "These savings are further increased by substantial reductions in drug acquisition costs, despite higher administration costs and costs for AEs. The total budget impact is estimated at a cost saving of £193.37 million over five years." by the toolchain. However, the reviewers noted a few occasions where the LLM had uncharacteristically poor phrasing, for example, using connecting words such as "although" erroneously and presenting results in an unintuitive order. On these occasions, the LLM updated the factual content of the statements while sticking rigidly to the phrasing of the original report. This hints at a limitation of the toolchain design (see Discussion).

Discussion

In HEOR, there are many repetitive, laborious tasks that could be performed by generative AI, including helping to conduct SLRs, network meta-analyses, economic modeling, and report writing.[10-12] Automating these tasks could help to free up time for the "critical thinking" and human-to-human aspects of the projects. Furthermore, it could help to reduce time and costs, ultimately expediting the HTA process and patient access to new treatments.[10,12] This is particularly pertinent in an era of unprecedented innovation, including the development of AI technologies in healthcare, which is outpacing the HTA process.[21]

A particular concern and challenge in all aspects of AI research is ensuring that the AI delivers at least human-level accuracy, if not better.[22] For healthcare applications, incorrect AI results can lead to patient harm, and in HEOR, it can lead to delays in access to treatment and be a waste of time and money. Therefore, it is vital to develop methods that deliver robust results. Here, we have developed LLM pipelines using techniques such as task decomposition, chain-of-thought prompting, and self-consistency prompting to achieve a high level of accuracy in automating parameter value adaptations and result reporting adaptations at a fraction of the time and costs compared with humans.

The pipelines were developed over the course of several months because significant experimentation was needed to identify optimal approaches. However, with the information provided in this article, researchers can replicate the pipelines in a much-reduced time frame and apply them to any number of adaptations.

Neither toolchain achieved perfect accuracy across all 3 examples. However, there are clear paths to improve each method. For the model adaptation pipeline, a clear improvement would be a conflict resolution step for step 3. If responses to the 2 ensemble-shuffled prompts result in conflicting values assigned to the same model cell, an additional LLM agent could be used to decide which assignment is more appropriate, based on the logic provided in each response. For the report adaptation toolchain, factual errors were primarily driven by a lack of understanding of HEOR terms. This limitation might be amenable to a dynamic context approach. A repository of HEOR terms could be included in the toolchain and dynamically matched to statements in the same way that the adaptation result repository is used by Tool 5. Because accuracy was not perfect, it is recommended that all outputs of the reported pipelines be reviewed by modeling experts, in the same way that manually adapted materials undergo review. There are opportunities to enhance this review process. For example, if a "conflict resolution" approach is used, any "conflict" cells could be automatically flagged in the model as areas of focus for review by the human health economist.

The study reported in this article has several limitations. First, a wider variety of examples would be beneficial to establish the generalizability of the toolchains and the methods used. Second, although the study investigates automated workflows, manual setup is required for both toolchains to function effectively. The time taken to perform this setup (once per global material) should be weighed against the cost and efficiency savings from automating adaptations. The pipelines would deliver the greatest value if the "AI-friendliness" of the materials were considered at an early stage in model and report development. However, it should be noted that most of the setup aligns with good modeling practice (eg, using a clearly labeled sensitivity analysis filter and including report tables and figures in the economic model). Third, we assessed the applicability of LLMs to 2 routine components of adaptation, parameter value adaptations and result reporting adaptations. We did not develop toolchains for adaptations to the introduction, methods, and input sections of reports or updates to Excel model engines. The central challenge for expansion to other report sections is handling context. For example, taking the input section, there is no immediately obvious source of context to update claims about why certain data sources were chosen over others. On the modeling side, novel approaches would be required to expand the scope of adaptations to include updates to the model engine. As of yet, no robust solution has been developed to enable LLMs to accurately interpret and edit complex chains of Excel formulae. Fourth, the model adaptation pipeline we described only updates deterministic parameter values. In theory, the same methods could be applied to update parameter distributions (used in probabilistic sensitivity analysis) if measures of variability (such as standard error) are provided alongside data points. However, this would require further research to confirm the capabilities of LLMs in distribution choice and calculation of distribution parameters.

In terms of further research, the authors hypothesize that incidental poor phrasing of the report adaptation toolchain might be caused by an overemphasis on retaining the original reports' word choice. For example, the LLM was noted on occasion to retain the wording of a sentence exactly, while only changing the numerical content, leading to a strangely laid-out statement. We emphasized retaining word choice because the messaging of reports is often crafted with careful attention. However, it might be possible to achieve an appropriate tone through other means that do not result in poor phrasing, such as fine-tuning or including tone guidelines in the prompting.

Conclusions

In conclusion, we developed 2 LLM-based toolchains that could have a high utility in automating the most common components of

health economic model adaptation. LLM-based approaches to automate model adaptation and other laborious HEOR methods could help to free up time for the "critical thinking" and human-to-human aspects of projects, as well as reduce time and costs, and ultimately expedite the HTA process and patient access to new treatments.

Author Disclosures

Author disclosure forms can be accessed below in the Supplemental Material section.

Supplemental Material

Supplementary data associated with this article can be found in the online version at https://doi.org/10.1016/j.jval.2025.05.020.

Article and Author Information

Accepted for Publication: May 29, 2025

Published Online: July 24, 2025

doi: https://doi.org/10.1016/j.jval.2025.05.020

Author Affiliations: Estima Scientific, London, England, UK (Rawlinson, Reason, Gimblett); Bristol Myers Squibb, Uxbridge, England, UK (Teitsson, Malcolm); Bristol Myers Squibb, Princeton, NJ, USA (Klijn).

Correspondence: William Rawlinson, MPhysPhil, Estima Scientific, 191 Wood Lane, London, England W12 7FP, United Kingdom. Email: will.rawlinson@estima-sci.com

Authorship Confirmation: All authors certify that they meet the ICMJE criteria for authorship.

Funding/Support: This study was supported by Bristol Myers Squibb.

Role of the Funder/Sponsor: The funder had no role in the design and conduct of the study; collection, management, analysis, and interpretation of the data; preparation, review, or approval of the manuscript; and decision to submit the manuscript for publication.

Acknowledgment: Amanda Prowse, PhD (Lochside Medical Communications), provided editorial support.

REFERENCES

1. Sehdev S, Gotfrit J, Elias M, Stein BD. Impact of systemic delays for patient access to oncology drugs on clinical, economic, and quality of life outcomes in Canada: a call to action. *Curr Oncol*. 2024;31(3):1460–1469.
2. Angelis A, Lange A, Kanavos P. Using health technology assessment to assess the value of new medicines: results of a systematic review and expert consultation across eight European countries. *Eur J Health Econ*. 2018;19(1):123–152.
3. Daniel Mullins C, Onwudiwe NC, Branco de Araújo GT, et al. Guidance document: global pharmacoeconomic model adaption strategies. *Value Health Reg Issues*. 2014;5:7–13.
4. Minaee S, Mikolov T, Nikzad N, et al. Large language models: a survey. Arxiv. https://arxiv.org/html/2402.06196v2; Published 2024. Accessed March 14, 2025.
5. Fleurence R, Wang X, Bian J, et al. Generative AI in health economics and outcomes research: a taxonomy of key definitions and emerging applications, an ISPOR working group report. Arxiv. https://arxiv.org/abs/2410.20204; Published 2024. Accessed March 17, 2025.
6. Fleurence RL, Bian J, Wang X, et al. Generative AI for health technology assessment: opportunities, challenges, and policy considerations-an ISPOR working group report. *Value Health*. 2025;28(2):175–183.
7. National Institute for Health and Care Excellence. Use of AI in evidence generation: NICE position statement. https://www.nice.org.uk/about/what-we-do/our-research-work/use-of-ai-in-evidence-generation–nice-position-statement; Published 2024. Accessed February 3, 2025.
8. Chhatwal J, Yildirim I, Balta D, et al. EE355 can large language models generate conceptual health economic models? *Value Health*. 2024;27(6):S123.
9. Poirrier JE, Bergemann R. MSR26 the use of copilot, a generative artificial intelligence tool, as programming assistant in excel-based health economic models. *Value Health*. 2023;26(12):S398.
10. Reason T, Benbow E, Langham J, Gimblett A, Klijn SL, Malcolm B. Artificial intelligence to automate network meta-analyses: four case studies to evaluate the potential application of large language models. *Pharmacoecon Open*. 2024;8(2):205–220.
11. Reason T, Langham J, Gimblett A. Automated mass extraction of over 680,000 PICOs from clinical study abstracts using generative AI: a proof-of-concept study. *Pharm Med*. 2024;38(5):365–372.
12. Reason T, Rawlinson W, Langham J, Gimblett A, Malcolm B, Klijn S. Artificial intelligence to automate health economic modelling: a case study to evaluate the potential application of large language models. *Pharmacoecon Open*. 2024;8(2):191–203.
13. Rawlinson W, Klijn S, Teitsson S, Malcolm B, Gimblett A, Reason T. P48 automating economic modelling: potential of generative AI for updating excel-based cost-effectiveness models. *Value Health*. 2024;27(6):S11.
14. Rawlinson W, Teitsson S, Reason T, et al. EE205 automating economic modeling: potential of generative AI for updating modeling reports. *Value Health*. 2024;27:S95.
15. Rawlinson W, Teitsson S, Reason T, Malcolm B, Gimblett A, Klijn S. HTA156 assessing the generalizability of automating adaptation of excel-based cost-effectiveness models using generative AI. *Value Health*. 2024;27(12):S384.
16. Khot T, Trivedi H, Finlayson M, et al. Decomposed prompting: a modular approach for solving complex tasks. ArXiv. https://arxiv.org/abs/2210.02406; Published 2022. Accessed March 26, 2025.
17. An C, Zhang J, Zhong M, et al. Why does the effective context length of LLMS fall short? ArXiv. https://arxiv.org/abs/2410.18745; Published 2024. Accessed March 26, 2025.
18. Zheng C, Zhou H, Meng F, Zhou J, Huang M. Large language models are not robust multiple choice selectors. ArXiv. https://arxiv.org/abs/2309.03882; Published 2023. Accessed March 26, 2025.
19. Wang X, Wei J, Schuurmans D, et al. Self-consistency improves chain of thought reasoning in language models. ArXiv. https://arxiv.org/abs/2203.11171; Published 2022. Accessed March 26, 2025.
20. Wang M, Chen L, Cheng F, et al. Leave no document behind: benchmarking long-context LLMS with extended multi-doc QA. AxXiv. https://arxiv.org/abs/2406.17419; Published 2024. Accessed March 27, 2025.
21. Vintura. Every day counts–improving time to patient access to innovative oncology therapies in Europe. https://www.efpia.eu/media/578013/every-day-counts.pdf; Published 2020. Accessed April 4, 2025.
22. Teno JM. Garbage in, Garbage out—Words of Caution on Big Data and Machine Learning in Medical Practice. *JAMA Health Forum*. 2023;4(2):e230397. e230397.