AIde Solutions

Augmenting Health Economic Model Report Creation with LLMs

barisdeniz@aidesolutions.net

Agenda

1. Foundation LLMs for use in research
2. Augmenting model report generation
3. Beyond report generation

Foundation LLMs for HEOR Research

Strengths

- **Rapid Information Processing**
  Quick synthesis of large volumes of text and efficient at summarizing content
- **Natural Language Understanding**
  Can interpret complex queries and context, and handle nuanced language
- **Versatility**
  Applicable across various HEOR activities and can adapt to different writing styles and formats
- **24/7 Availability**
  Consistent performance without fatigue and immediate responses to queries
- **Broad Knowledge Base**
  Trained on diverse datasets, including medical literature, providing interdisciplinary insights

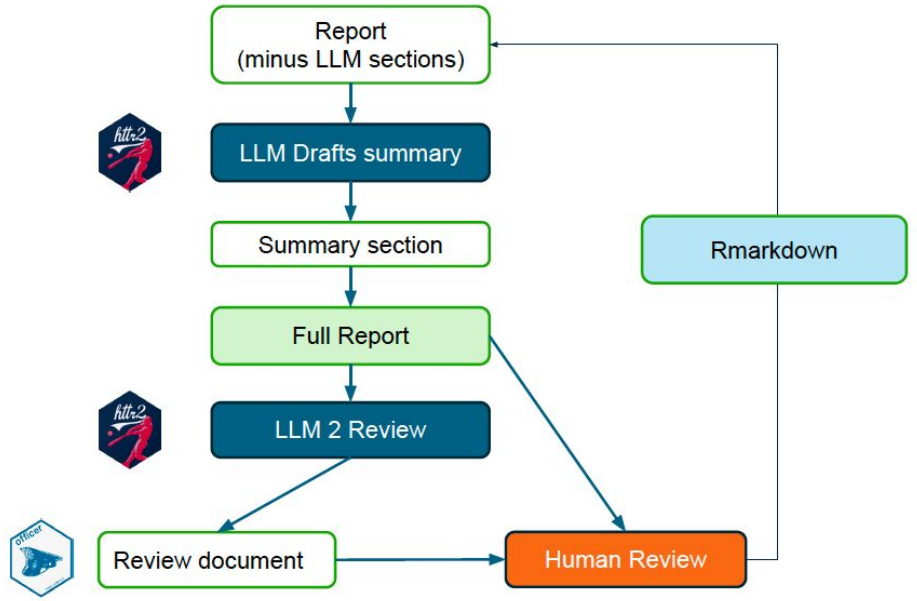
Limitations

- **Not a method expert**
  No genuine comprehension of scientific concepts
- **Hallucinations**
  Can generate plausible-sounding but incorrect information, with a risk of 'hallucinations' or fabricated data
- **Absence of Critical Thinking**
  Cannot independently evaluate the quality of information or form novel scientific insights
- **Multi-step Tasks**
  May struggle with complex, multi-step tasks
- **Relevant Data**
  Knowledge cutoff date limits access to recent developments and may lack specific, niche HEOR knowledge

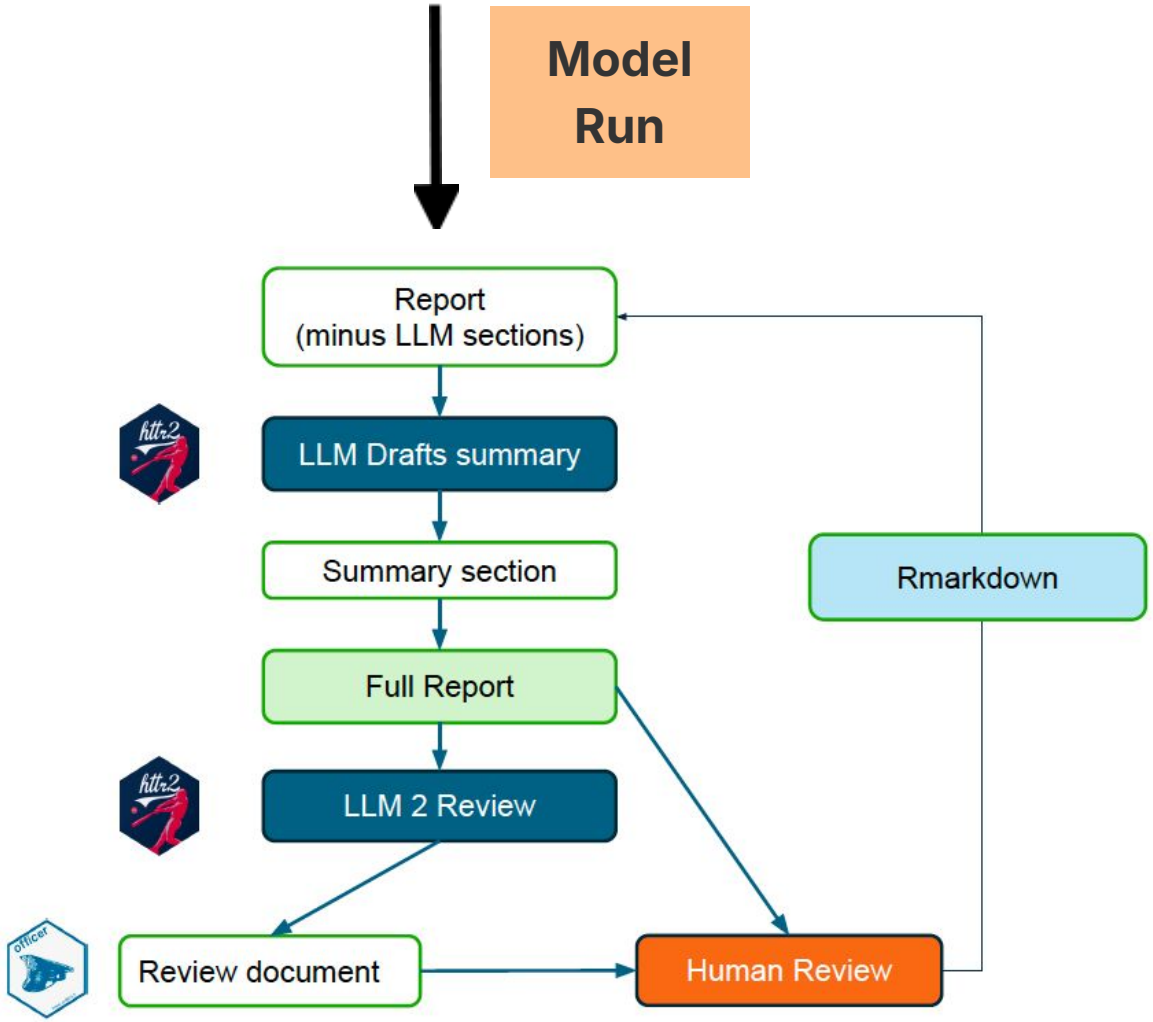
(Obvious Application): Report Augmentation

Large Language Models (LLMs) can augment traditional report generation by providing dynamic, contextual narrative sections.

LLMs excel at generating high-quality, coherent text that interprets data, summarizes key insights, and provides tailored conclusions - capabilities beyond the scope of static R Markdown templates.

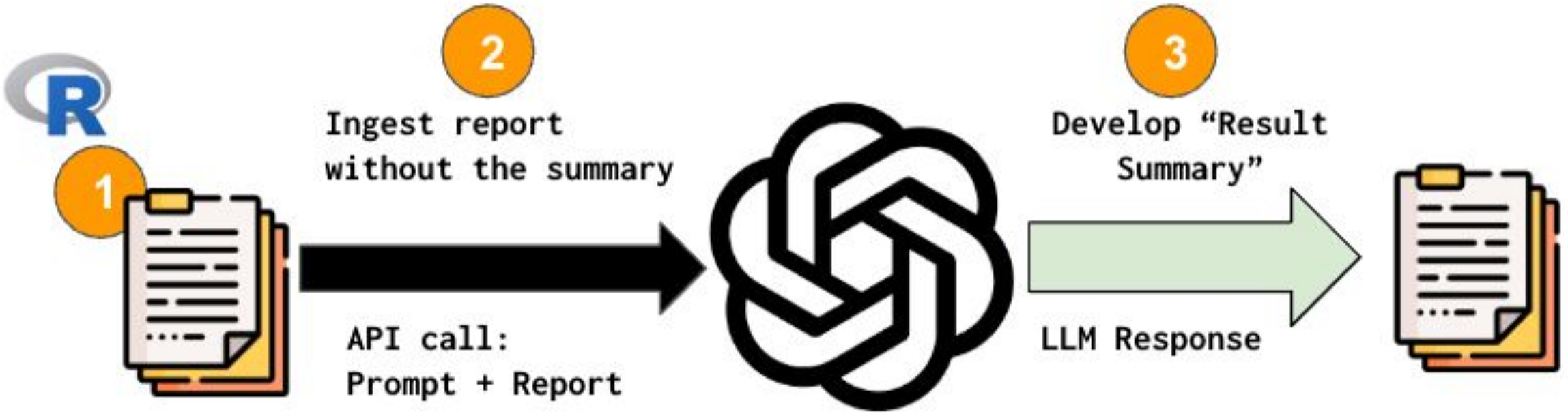


From Rob Smith's presentation



1. Run the model
2. Generate "static" part of the report (Rmarkdown)
3. LLM call for result summary creation
4. LLM call for report review

Overview of LLM call from R



```r
writeLLMsummary <- function(
    input_path,
    output_path,
    base_url = "https://generativelanguage.googleapis.com/v1beta/models/",
    model = "gemini-1.5-flash",
    n_tokens = 30000) {
  # Extract text from the Word document
  text <- extract_text_from_word(input_path, n_tokens)
  summary <- query_llm(
    text = text,
    base_url = base_url,
    model = model,
    prompt = "Write the summary section of this document in prose in under 200 words"
  )
  writeLines(
    text = summary,
    con = output_path
  )
  cat(paste0("LLM summary section saved to: ", output_path), "\n")
}
```

Function definition
API address
LLM Model
Size of the report to be extracted

"Context"
Static Report (30000 tokens)

LLM Call
Context
Prompt

LLM Response

```r
query_llm <- function(text,
                      base_url,
                      model,
                      prompt) {
  full_url <- paste0(base_url, model, ":generateContent")

  body <- jsonlite::toJSON(list(
    contents = list(
      list(
        parts = list(
          list(text = paste0(prompt, ":\n\n", text))
        )
      )
    )
  ), auto_unbox = TRUE)

  response <- httr::POST(
    url = paste0(full_url, "?key=", Sys.getenv("GEMINI_API_KEY")),
    httr::content_type_json(),
    body = body
  )

  result <- httr::content(response, as = "text", encoding = "UTF-8")
  json <- jsonlite::fromJSON(result)
  return(json$candidates$content$parts[[1]]$text)
}
```

Function definition

"text": static report

"prompt": instruction to LLM

"Context"

Converts R list (the "context") to JSON format

Combines the "prompt" and the "text"

LLM Request

Communicate with LLM using **httr** package and **Gemini Api Key**

Shares the "context"

LLM Response

Receives LLM output

Extracts the output in text

Foundation LLMs vs LLM as Research Tools

Guardrails

Appropriate Context

Audience

Example

Traceability

Milestone

Increasing Complexity in LLM System Design

Basic Interaction	Chains of Thought / Sequential Workflows	Conversational / Iterative Systems	Parallel / Independent Multi-Agent Systems	Collective / Collaborative Multi-Agent Systems	Self-Improving / Adaptive Systems
A single input prompt and a single output response, the simplest form of LLM interaction.	LLM outputs from one step become inputs for the next, enabling multi-step reasoning, task decomposition, and integration of external tools.	Back-and-forth interaction with a user or another system, maintaining context over multiple turns for refinement and clarification.	Multiple LLM agents operating concurrently on distinct tasks, with their outputs aggregated or presented separately.	Multiple LLM agents with defined roles working together on a common goal, communicating, debating, and iterating on solutions collaboratively.	Systems that learn from their own performance, user feedback, or external data to refine their prompts, tool usage, or internal reasoning processes over time.

Augmenting Health Economic Model Report Creation with LLMs

The use of Large Language Models (LLMs) presents a promising opportunity to enhance the creation of health economic model reports and other HEOR tasks. By leveraging the capabilities of LLMs, organizations can streamline report generation, improve consistency, and free up valuable time for higher-level analysis and decision-making.