Using LLMs to Generate Materials, Individualize Participant Experiences, and Role-Play in Studies

Richard N. Landers

CARMA Webcast Lecture Series
April 17, 2026

University of Minnesota
Driven to Discover®

This Workshop

- LLMs are being adopted quickly in research
- Guidance is thin, creating hidden internal and construct validity threats
- Social scientists are not generally trained in the engineering practices needed to do this well

First Principles

- Do ***not*** use LLMs if you don't need to.
  - Validated measures already exist.
  - Previously used participant materials have already been vetted.

- Do ***not*** use more complex LLM approaches if you don't need to.
  - More complexity creates more validity threats to manage.

Ethical Considerations

- **Privacy**
  - Participants want assurances. You need to understand what's realistic/correct.
- **Safety**
  - You must engineer safety yourself. Consider hate speech, interaction harm, misinformation, and malicious use.
- **Bias**
  - Training data misrepresent populations and essentialize demographics.
- **Broader Concerns**
  - Energy costs, copyright concerns, ghost workforce

Understanding What LLMs Are

- What comes next: **HIP HIP ???**

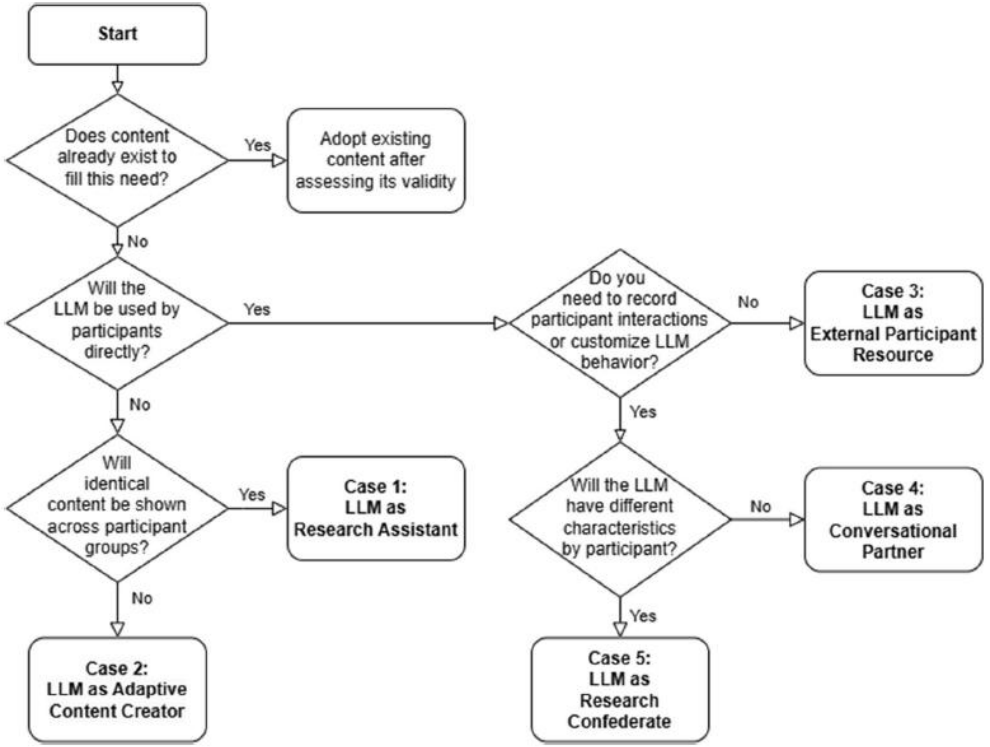
- LLMs are ML-based (text) prediction models!
- This is a **piece** of human thought and reasoning.
- Just this **piece** is **impressive and powerful**.
  - Because LLMs are better at this **piece** than humans, humans tend to assume they are "better" overall.

- The *model* is different from the *interface*.
  - GPT-*n* vs. ChatGPT, Haiku/Sonnet/Opus vs. Claude.com

Decision Tree

- What are you trying to accomplish?

- Behrend, T. S. & Landers, R. N. (2026). Participant Interactions with Artificial Intelligence: Using Large Language Models to Generate Research Materials for Surveys and Experiments. *Journal of Business and Psychology, 40,* 1275-1297.



Cases 1 & 2: Content Generation

- **Case 1: Research Assistant**
  - You've probably done this already using ChatGPT, Gemini, or Claude…
- **Case 2: Adaptive Content Creator**
  - Unique stimulus materials are shown to different participants.
  - Simulates a study proctor modifying materials in real-time.
  - For Qualtrics integration, use **Content QUAIL**

Cases 3, 4 and 5: Participant Interaction

- **Case 3**: External Resource
  - Send the participant to ChatGPT! (*not recommended*)

- **Case 4/5:** Conversational Partner
  - *Case 4:* Simulate a conversation with a study proctor/actor
  - *Case 5:* Assign people to different conversational prototypes
  - For Qualtrics integration, use **Conversational QUAIL**

Relevant Tools for Demos Today

- **Required**
  - Qualtrics, with JavaScript access
  - LLM access (we will use GPT-*n*)

- **Optional**, for development/testing
  - R, R Studio

Demo 1: Getting GPT-n Access

- Set up your API account
  - https://platform.openai.com
- Fund your account
  - $10 is plenty for testing, most studies will be < $50
- Generate and save your API key
  - API keys left side menu → Create new secret key → Restricted → Model capabilities → Chat completions / Request → Responses / Write → Create secret key

Prompt Engineering as Engineering

- Prompt engineering is an engineering process used to develop the model's **system prompt**
- Engineering is not science
- Engineering requires a different mindset
  - Iteration toward concrete goals
  - Incremental improvement until "good enough"

Two Frameworks to Guide Engineering

- **Design thinking**
  - Empathize → Define → Ideate → Prototype → Test
  - Return to any earlier stage whenever you are struggling at a later stage

- Test driven development using **red-green-refactor**
  - Red = identify test cases that may fail and test them
  - Green = only make additional changes once the test is passed
  - Refactor = return to define, ideate, or prototype

Empathize ↔ Define

- **Empathize**: What do you want participants to experience? How will they experience it?
  - Case 4 ex.: Leadership coach

- **Define**: Develop metrics or test cases (as appropriate) to achieving your empathized vision.
  - No offensive conversations
  - No easy answers

Ideate

- Brainstorm potential prompting approaches
- There is no correct answer

Table 2 Taxonomy of design dimensions for LLM selection and system prompt design, adapted through the integration of Braun et al. (2024) and Johns (2006)

Dimension	Characteristics	Explanation
Dimensions affecting prompt design alone		
Interaction type	Researcher-in-the-loop Participant-in-the-loop Fully autonomous	Case 1 represents a researcher-in-the-loop model, where the researcher can moderate all outputs. Case 2 is fully autonomous, with participants acting as passive recipients of LLM-generated output. Cases 3–5 correspond to participant-in-the-loop model, where participant actions can influence future LLM actions within the same conversation. Prompts should be written to make interaction type explicit, e.g., "You are a question-answering assistant." versus "You are trying to help the user address a problem on their own.".
Goal	Learn Lookup Investigate monitor/extract Decide Create	LLMs should always be assigned an explicit purpose or goal, as this more effectively guides their behavior, particularly when user input deviates from expected conversational parameters. The list of characteristics provided serves as examples only; goals may be represented by any action words. However, different action words, even those apparently synonymous, can result in markedly different interactions and therefore require deliberate prompt engineering.
Role	Defined omnibus context Defined discrete context Fully defined context Undefined	LLMs operating with undefined roles generate predictions based on default prompts or latent, unobservable rules. For instance, ChatGPT uses an existing system prompt, hidden from users, that guides its behavior separately from that of the GPT-n algorithm. For research purposes, undefined roles are generally undesirable. Instead, researchers should define the role of the LLM explicitly, within both an omnibus and discrete context (Johns, 2006). This might require, for example, telling the LLM its occupation and purpose (i.e., omnibus context), alongside a specific task to accomplish and the social context in which to do so (i.e., discrete context).
Style	Defined Undefined	Beyond the specific role of the LLM, instructions can be provided to encourage it to act in specific ways. Such styles should be informed by validated psychological frameworks, such as by giving a personality framework or describing specific attitudes, norms, or values.
Dimensions affecting both LLM choice and prompt design		
Learning	Zero-shot One-shot Few-shot Many-shot	Simply providing direct instructions in the prompt (i.e., a "zero-shot" approach) may be insufficient to produce desired interactions. Other times, it may be useful to provide one ("one-shot") or more ("few-shot") examples within the prompt of what a "good" interaction might look like. This can range from a single conversational turn to a long series of turns, or even one or more entire example conversations. Additionally, many LLM providers offer the ability to fine-tune models, referring to the provision of large datasets (i.e., a "many-shot" approach) to more fundamentally customize the underlying model for a specific purpose. Rather than prompt engineering, this results in the creation of a bespoke model. Most organizational researchers will not have suitable pre-existing datasets for this purpose, making this an uncommon approach. However, the development of foundational datasets to be shared among researchers for this purpose could be a valuable research goal within particular domains.
Chain of thought	Single-step prompt Step-by-step prompt	Prompts usually contain relatively simple directives to shape LLM output. However, if the conversation is expected to be relatively complex, the prompt may explicitly name a series of steps to

Example After Initial Ideation

- You are a helpful friend named Ann Marie, trying to guide the user in identifying a productive short-term goal that will help the user reach their long-term goals. The user is talking to you because they need help, but you don’t know the user’s long-term goal yet. You should start the conversation by asking about it. Once you know what the long-term goal is, provide two concrete suggestions and ask the user which they prefer. Be supportive but critical.

Demo: Prototyping

- Easier to develop a system prompt in R than in Qualtrics

- You need to determine:
  - Model
  - Hyperparameters (temperature)
  - System prompt
  - Initial conversational turn

Prototyping ↔ Testing

- You will make 80% of engineering progress in the first 20% of the time you spend.

- Remember **red-green-refactor.**
  - You will uncover new failure conditions as you prototype.
  - These become new test cases.
  - Any change necessitates re-running **all test cases**.

Demo: Moving Case 4 into Qualtrics

Conversational QUAIL: https://osf.io/6ypgu/files/9m2ar

1. Create a Text/Graphic object
2. Change the text to whatever prompt you'd like to appear above the conversational interface.
3. Click **JavaScript** under Question behavior left panel
4. Copy/paste the code from OSF into this window.
5. Change the values as appropriate.
6. Click **Save**.
7. Open the Survey flow and add embedded data matching your settings.
8. Publish and test!

Testing → Launch

- Your API key is exposed client-side. Understand the risks and manage them.
  - Never fund your account more than you're willing to lose.
- Batched launches (with key updates) are recommended for big data collection efforts.
- Long-running projects will need active monitoring for participant safety.

Limitations to Remember

- Conversational success (i.e., internal validity) will only be as good as your engineering process. Use beta tests.
- LLMs are not human-simulators and do not "reason" like humans do. They have no affect. Participants may differ.
- Conversations are not reproducible unless temperature is set to zero, but this will affect flow.
- Models will "hallucinate," but remember what that means.

Takeaways

1. Use LLMs only when necessary for your RQs.
2. Prompt engineering is iterative, empirical, and requires good engineering practices (design thinking + red-green-refactor).
3. Research expertise, not the LLM itself, is the cause of internal/construct validity. Pilot/beta tests are strongly recommended.

Thank You!

Richard N. Landers
Department of Psychology

CARMA Webcast Lecture Series
April 17, 2026

Contact info, PDFs,
OSF and other links:

rlanders@tntlab.org

https://rlanders.net

UNIVERSITY OF MINNESOTA
Driven to Discover®