Micro-Prompting LLMs and SLMs

From Copilots to Agentic Workloads

Donald Thompson
Distinguished Engineer
Microsoft / LinkedIn

Agenda

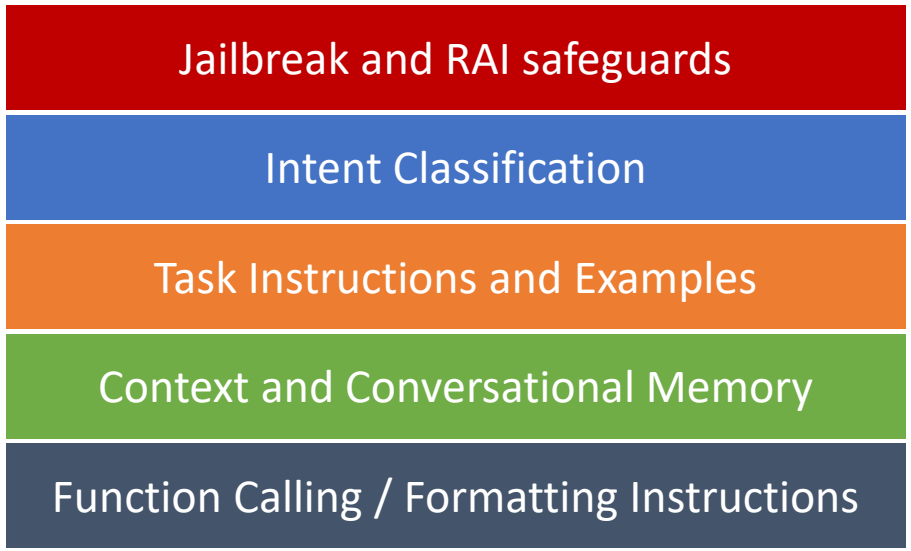
- Macro-prompting vs micro-prompting
- Micro-prompting example
- Automatic prompt optimization
- Automatic fine-tuning
- SLMs vs LLMs

Macro-Prompting

The Current Paradigm in GenAI

- Dominant approach since late 2022
- Crafting extensive, detailed prompts
- Provides comprehensive instructions to frontier LLMs

Anatomy of a Macro-Prompt

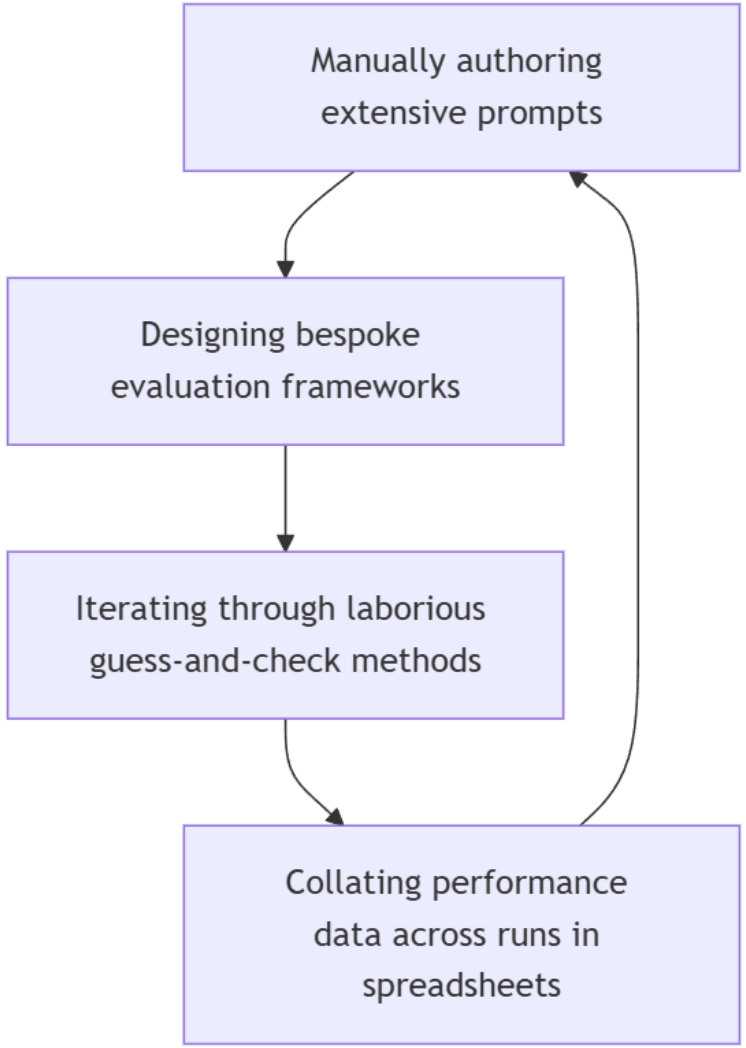


> 8k tokens

Macro-Prompting

Application Engineering Challenges

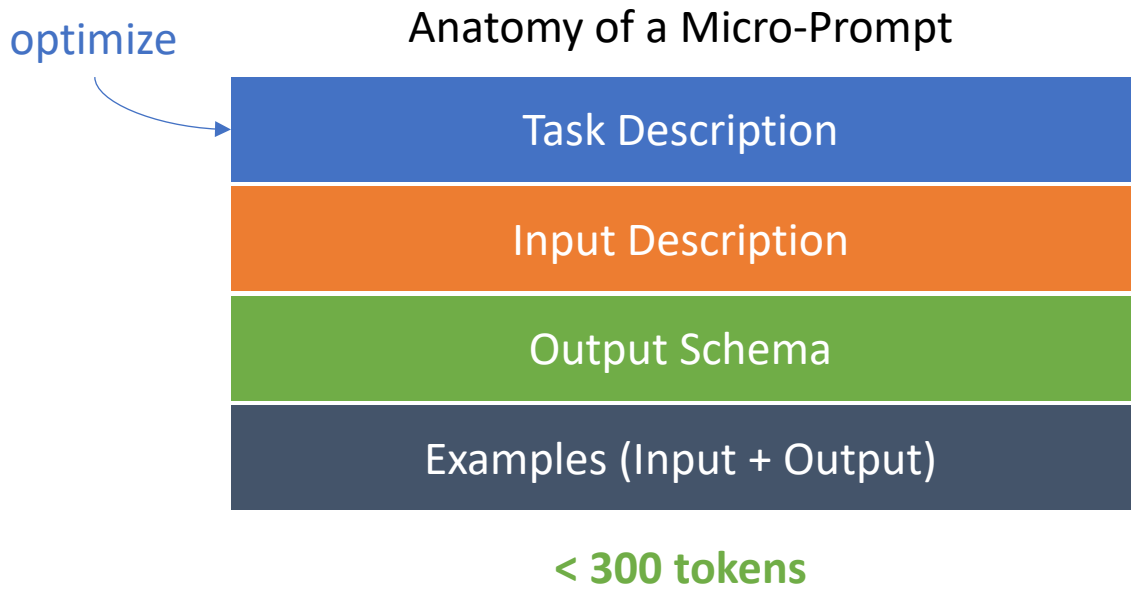
- Limited expertise in prompt engineering
- Limited time budget for iteration and evaluation
- Default reliance on expensive, scarce models (GPT-4)
- Reduced opportunity for fine-tuning



Micro-Prompting

A Paradigm Shift in GenAI Development

- Automated
- Goal-oriented, measurable
- Concise, modular, highly optimized
- Cost-effective on commodity h/w
- Scalable, durable process



Micro-Prompting

A New Workflow

- Decompose problems into discrete functional tasks
- Clearly-defined inputs and outputs
- Synthesize examples (input + output)
- Define measurement criteria

Modular Prompt Architecture

Module	Sample
Task	Propose the top 3 best seats for a traveler with the provided seat preferences, given the list of available seats.
Input 1	“Seat Preferences”: The seating preference for the traveler.
Input 2	“Available Seats”: An array of seats, each in the format [class]_[seat]_[aisle|middle|window]
Output	{ "type": "object", "properties": { "recommended_seats": { "description": ".", "type": "array", "items": { "type": "string" } } }, "required": ["recommended_seats"] }
Example	# INPUT Seat Preference: I usually fly business class. Window seats are preferred, but aisle is ok too. If I have to fly economy, it must be an exit row! Available Seats: [“First_3A_Aisle”, “Econ_22C_Middle”, “PremEcon_13F_Window”] # OUTPUT { “recommended_seats”: [“First_3A_Aisle”] }

Optimization

Dataset and Evaluation

- Example (training) datasets (input + output)
- Evaluation (loss) function

$$f : (\text{example}, \text{output}) \mapsto \text{score}$$

1. Calculate the Jaccard similarity
2. Convert similarity to loss (1 - similarity)
3. Add penalties for wrong seat numbers and invalid seats
4. Combine into a final score.

Perfect prediction results in a loss of 0; errors increase the loss, potentially above 1.

```
You are a synthetic dataset generator. The following is a task description for a machine learning model. Using this description, generate a dataset with a good variety of inputs and outputs that can be used to train, test, and validate the model:

```
{{task}}

{{inputs}}

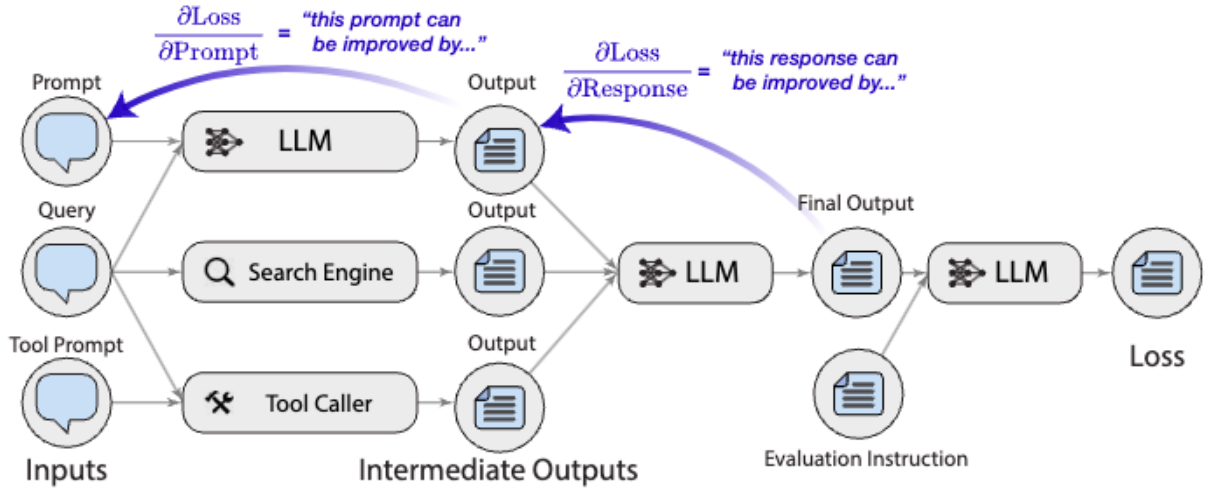
{{outputs}}

{{example}}
```

Your output should be in the following format:

[{"input": ..., "output": ...}, ...]
```

Optimization

Reverse Mode Automatic Differentiation

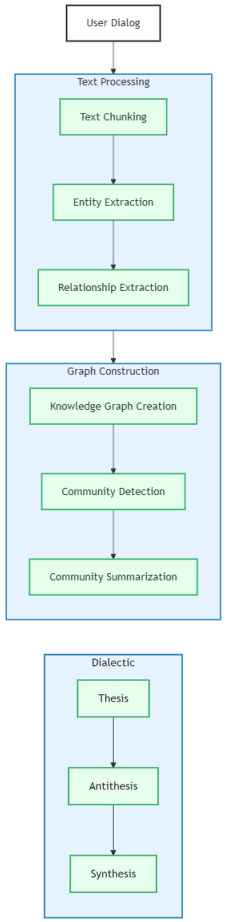
- Forward propagation
  - (model, prompt) being "trained" is given input and produces output
  - Use "weaker"/"cheaper" model
- Loss calculation
  - Evaluates the model's performance
  - LLM (frontier) as judge
- Textual gradients
  - LLM (frontier) provides feedback to improve the prompt
- Backpropagation
  - LLM (frontier) edits prompt (applies gradient)

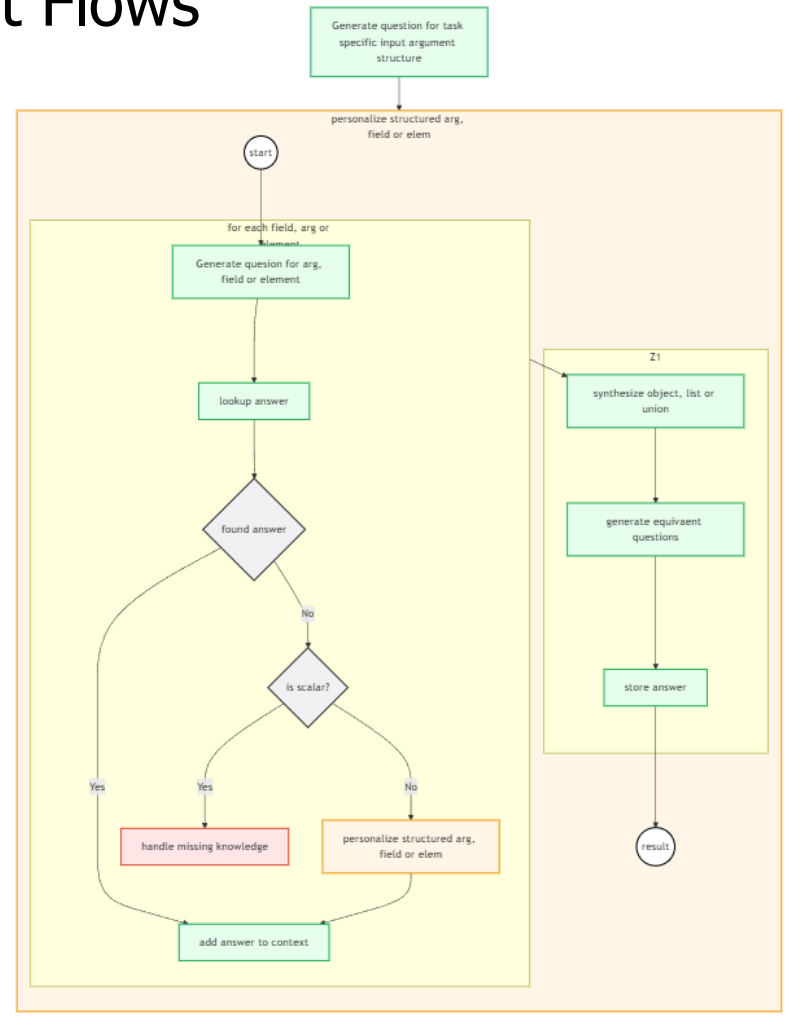


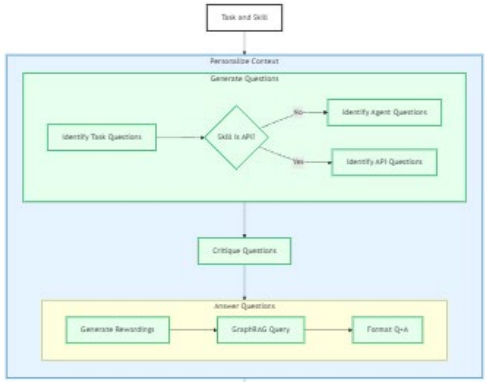
Yuksekgonul et al. (2024). TextGrad: Automatic "Differentiation" via Text. arXiv:2406.07496.

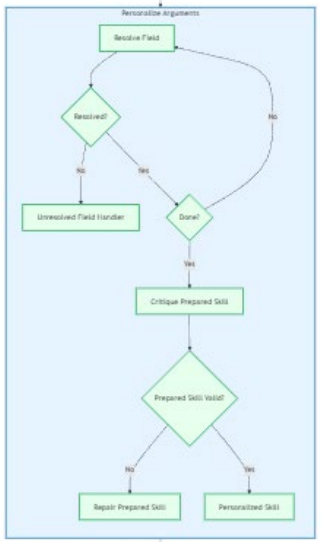
Agentic Workloads

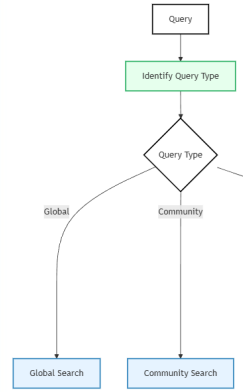
Micro-prompt Flows

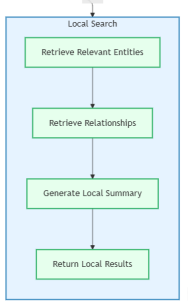












Automatic Fine-Tuning

Experimental

- Instruction fine-tuning datasets already defined
  - Instruction = optimized prompt
  - Input and outputs
- Combine for common flows into larger datasets
- Online re-optimization based on loss function monitoring

LLMs vs SLMs

Tradeoffs

LLMs

- Powerful but resource intensive
- General purpose
- High operational costs
- Data privacy, RLHF, and other concerns
- Limited customization / control
- Low barrier of entry

SLMs

- Efficient for micro-prompts
- Cost effective to deploy
- Enhanced data privacy and control
- Highly customizable
- Faster inference (potentially)
- Requires more technical expertise