Scaling Up "Vibe Checks" for Large Language Models

Shreya Shankar
April 2024

EPIC DATA lab
UC Berkeley

LLM Pipelines

- "Zero-shot" capabilities of LLMs enable intelligent data processing pipelines *without training models*

julia/podcaster-tweet-thread

Take a podcast episode transcript and turn into a tweet thread.

{×} Prompt • Updated a day ago • 8 • 866 • 107 • 6

homanp/github-code-reviews

This prompt reviews pull request on GitHub.

{×} Prompt • Updated 7 months ago • 12 • 3.62k • 451 • 8

matu/customer_satisfaction

This prompt is being use to extract services and sentiments from a customer answer to a survey (specially 1 question, How can we improve?)

{×} Prompt • Updated 6 months ago • 3 • 611 • 109 • 1

muhsinbashir/youtube-transcript-to-article

Convert any Youtube Video Transcript into an Article (SEO friendly)

{×} Prompt • Updated 6 months ago • 43 • 9.14k • 10.3k • 1

LangSmith

LLM Pipelines

- "Zero-shot" capabilities of LLMs enable intelligent data processing pipelines *without training models*

YouTube

Input Document

[00:02] welcome to this documentary on the making of the music video for "Driver's License" by…

Prompt Template

Your task is to take a given YouTube transcript and transform it into a well-structured and engaging article. Your objectives are as follows…

LLM(s)

Output

In this behind-the-scenes look at Olivia Rodrigo's hit music video…

LLMs Make Unpredictable Mistakes

- Hallucinations, bad formatting, ignoring instructions, & more.

YouTube

Input Document

[00:02] welcome to this documentary on the making of the music video for "Driver's License" by…

Prompt Template

Your task is to take a given YouTube transcript and transform it into a HTML-structured engaging article… avoid copying sentences directly…

LLM(s)

Output

<p> In this behind-the-scenes look… </div>

LLMs Make Unpredictable Mistakes

- Hallucinations, bad formatting, ignoring instructions, & more.

YouTube

Input Document

[08:14] yeah, I love it here. There's just so much inspiration in this…

Prompt Template

Your task is to take a given YouTube transcript and transform it into a HTML-structured engaging article… avoid copying sentences directly…

LLM(s)

Output

…There is just so much inspiration…

Vibe Checks, Rules, and Guardrails

- People rely on rules & guardrails to improve accuracy in traditional ML pipelines
- Hard to do for LLMs
  - What does "accuracy" mean for free-form text?
  - Metrics might be complicated, requiring humans or LLMs to evaluate

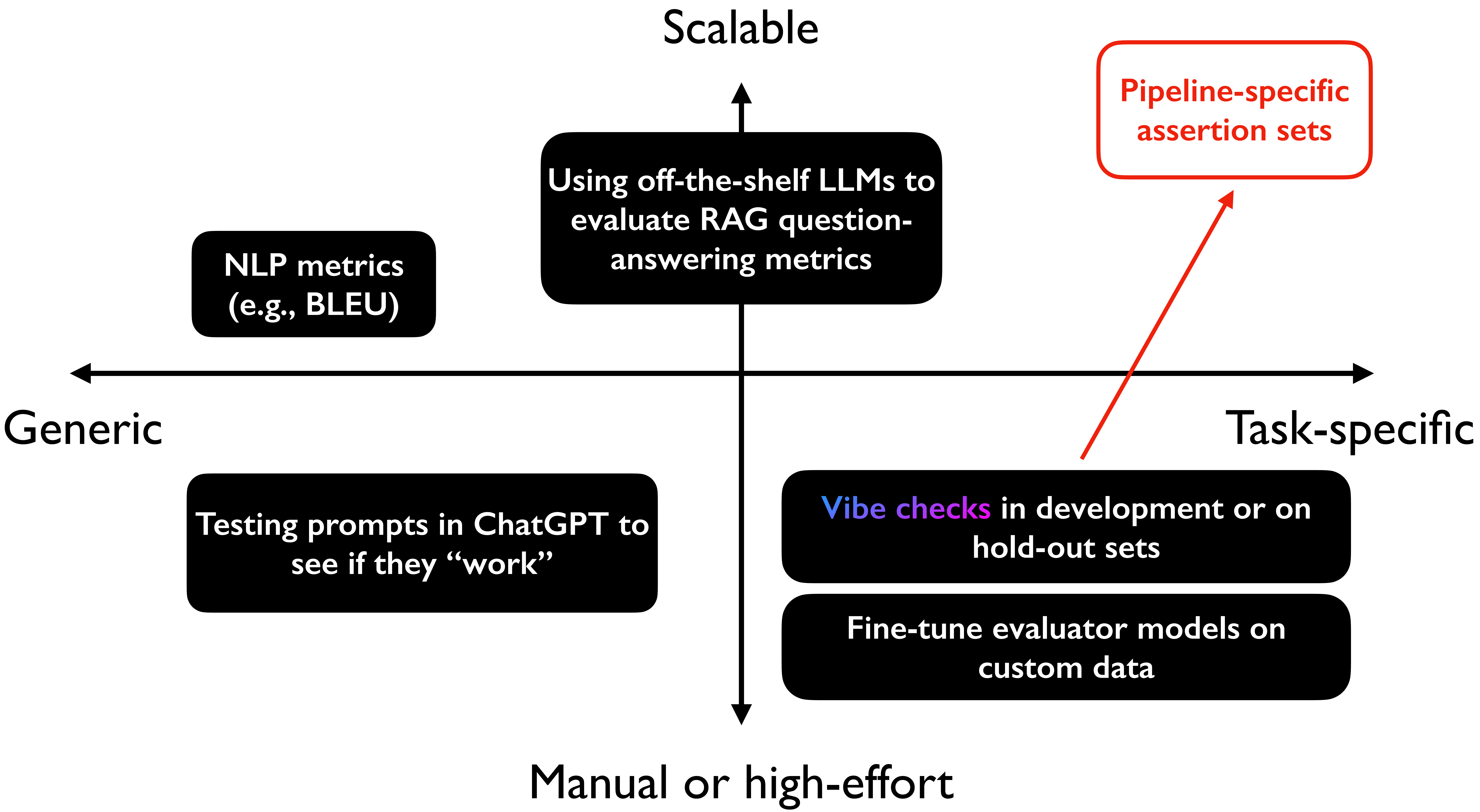
NLP metrics (e.g., BLEU)

RAG question-answering metrics (e.g., faithfulness, relevance, context recall)

Vibe checks

Generic ⟷ Task-specific

Vibe Checks, Rules, and Guardrails



Evaluation Assistants

- *Evaluation assistants:* tools that aid humans in creating **task-specific evaluations and assertions** that align with how they would grade pipeline outputs
- Today's talk:
  - Auto-generating criteria and assertions
  - Insights from large-scale deployment with LangChain
  - Mixed-initiative interface to develop custom assertions
  - Lessons learned from small-scale qualitative study

Shankar, Shreya, Haotian Li, Parth Asawa, Madelon Hulsebos, Yiming Lin, J. D. Zamfirescu-Pereira, Harrison Chase, Will Fu-Hinthorn, Aditya G. Parameswaran, and Eugene Wu. "SPADE: Synthesizing Data Quality Assertions for Large Language Model Pipelines." *Under submission.*

Shankar, Shreya, J. D. Zamfirescu-Pereira, Bjoern Hartmann, Aditya G. Parameswaran, and Ian Arawjo. "Who Validates The Validators? Aligning LLM-Assisted Evaluation of LLM Outputs with Human Preferences." *Under submission.*

Evaluation Assistants

- *Evaluation assistants:* tools that aid humans in creating **task-specific evaluations and assertions** that align with how they would grade pipeline outputs
- Today's talk:
  - Auto-generating criteria and assertions
  - Insights from large-scale deployment with LangChain
  - Mixed-initiative interface to develop custom assertions
  - Lessons learned from small-scale qualitative study

Shankar, Shreya, Haotian Li, Parth Asawa, Madelon Hulsebos, Yiming Lin, J. D. Zamfirescu-Pereira, Harrison Chase, Will Fu-Hinthorn, Aditya G. Parameswaran, and Eugene Wu. "SPADE: Synthesizing Data Quality Assertions for Large Language Model Pipelines." *Under submission.*

Shankar, Shreya, J. D. Zamfirescu-Pereira, Bjoern Hartmann, Aditya G. Parameswaran, and Ian Arawjo. "Who Validates The Validators? Aligning LLM-Assisted Evaluation of LLM Outputs with Human Preferences." *Under submission.*

Auto-Generated Assertions

"*Summarize this document {doc_text}. Return your answer in markdown. Don't include any sensitive information like race or gender. Have a professional tone.*"

12/9/2018

Patient Medical Record

Patient Information | Birth Date

LLM Output	Is markdown	Doesn't include sensitive information	Has Professional Tone
"# Medical History\nThis document describes someone's medical history…"	✅	✅	✅
"# Medical History\n this describes shreya shankar's medical history while living in a fun neighborhood in SF…"	✅	❌	❌
"# Medical History\nThis describes Shreya Shankar's medical history while living in San Francisco…"	✅	❌	✅
"I'm sorry, but as a language model trained by OpenAI…"	❌	✅	✅

Need coding experience to write

Hard to evaluate. Need LLM?

Shankar, Shreya, Haotian Li, Parth Asawa, Madelon Hulsebos, Yiming Lin, J. D. Zamfirescu-Pereira, Harrison Chase, Will Fu-Hinthorn, Aditya G. Parameswaran, and Eugene Wu. "SPADE: Synthesizing Data Quality Assertions for Large Language Model Pipelines." *Under submission.*

Generating Assertions: Overview

- Goal: generate a minimal set of assertions with good coverage of failures and good accuracy
- Challenges:
  - How can we find the assertion functions desired by the developer?
  - How should we guarantee the coverage of failures with minimum # of assertions?
- SPADE (**S**ystem for **P**rompt **A**nalysis and **D**elta-Based **E**valuation) employs a two-stage workflow including (1) *generating candidate assertions* and then (2) *filtering candidate assertions*.

Shankar, Shreya, Haotian Li, Parth Asawa, Madelon Hulsebos, Yiming Lin, J. D. Zamfirescu-Pereira, Harrison Chase, Will Fu-Hinthorn, Aditya G. Parameswaran, and Eugene Wu. "SPADE: Synthesizing Data Quality Assertions for Large Language Model Pipelines." *Under submission.*

Generating Assertion Criteria

Criteria are hidden in prompt version histories!

Summarize this document {doc_text}. Return your answer in markdown.

↓

Summarize this document {doc_text}. Return your answer in markdown. If the document has sensitive information, don't include it in the summary.

↓

Summarize this document {doc_text}. Return your answer in markdown. ~~If the document has sensitive information, don't include it in the summary~~. DO NOT under any circumstances include sensitive information (e.g., race, ethnicity, gender).

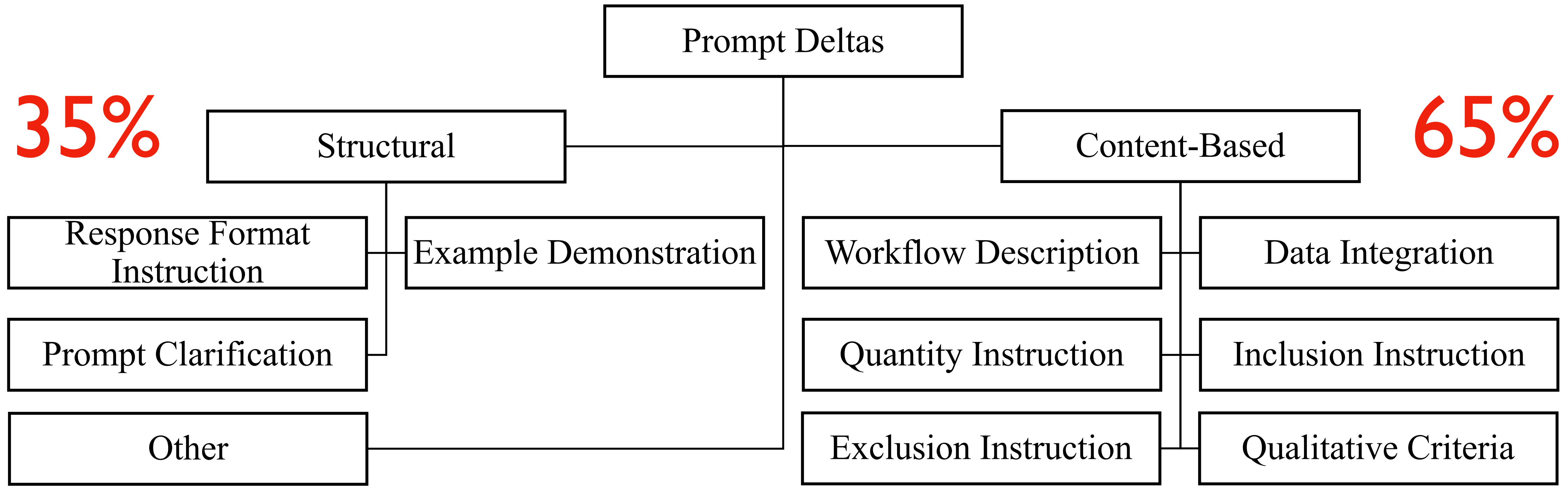
↓

Summarize this document {doc_text}. Return your answer in markdown. ~~DO NOT under any circumstances include sensitive information (e.g., race, ethnicity, gender)~~. Don't include any sensitive information like race or gender. Have a professional tone.

Shankar, Shreya, Haotian Li, Parth Asawa, Madelon Hulsebos, Yiming Lin, J. D. Zamfirescu-Pereira, Harrison Chase, Will Fu-Hinthorn, Aditya G. Parameswaran, and Eugene Wu. "SPADE: Synthesizing Data Quality Assertions for Large Language Model Pipelines." *Under submission.*

Categorizing Prompt Deltas to Inform Assertions

Across 19 LLM pipelines…



Category	Example Addition or Edit to a Prompt	Assertion Criteria
Response Format Instruction	*“Return your answer in Markdown”*	Parse to markdown correctly
Example Demonstration	*“Here is an example summary: # Medical History…”*	Infer detailed structure from example
Prompt Clarification	*“~~Return~~ Give me a descriptive answer”*	N/A
Workflow Description	*“First, check for any tables or images. Then, …”*	Check for table summaries
Data Integration	*”The document info is {doc_info}”*	N/A
Quantity Instruction	*“The response should be at least 100 words”*	> 100 words
Inclusion Instruction	*“The title should be the same and end in Summary`”*	Assert same title + “Summary”
Exclusion Instruction	*“Do not include sensitive information”*	No name, race, gender, etc.
Qualitative Criteria	*“Your response should be in a professional tone”*	Professional tone

Shankar, Shreya, Haotian Li, Parth Asawa, Madelon Hulsebos, Yiming Lin, J. D. Zamfirescu-Pereira, Harrison Chase, Will Fu-Hinthorn, Aditya G. Parameswaran, and Eugene Wu. "SPADE: Synthesizing Data Quality Assertions for Large Language Model Pipelines.” *Under submission.*

From Taxonomy to Candidate Assertions

"- DO NOT under any circumstances include sensitive information (e.g., race, ethnicity, gender). + Don't include any sensitive information like race or gender. Have a professional tone."

→

Criteria	Category	Source
No sensitive information	Exclusion	"Don't include any sensitive information like…"
Professional tone	Qualitative Criteria	"Have a professional tone"

Find *as many as possible*

→

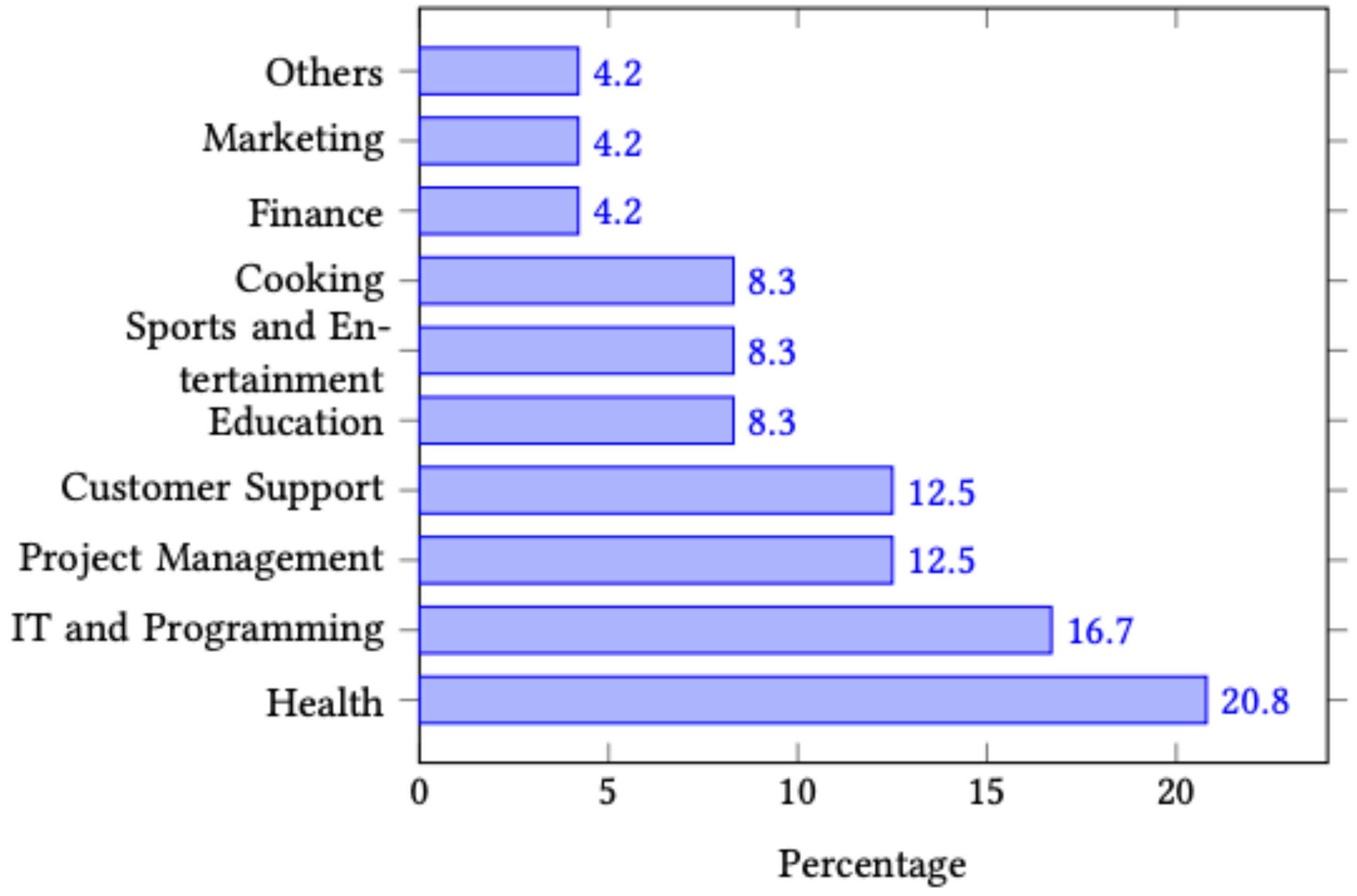
```
def assert_sensitive_1(prompt,
response):
  return "race" not in response and
  "gender" not in response

def assert_sensitive_2(prompt,
response):
  return "male" not in response and
  "female" not in response

def assert_sensitive_3(prompt,
response):
  return ask_llm(f"Is there sensitive
  information like race or gender in
  {response}?")

def assert_prof(prompt, response):
  return ask_llm(f"Is the tone here
  professional: {response}?")
```

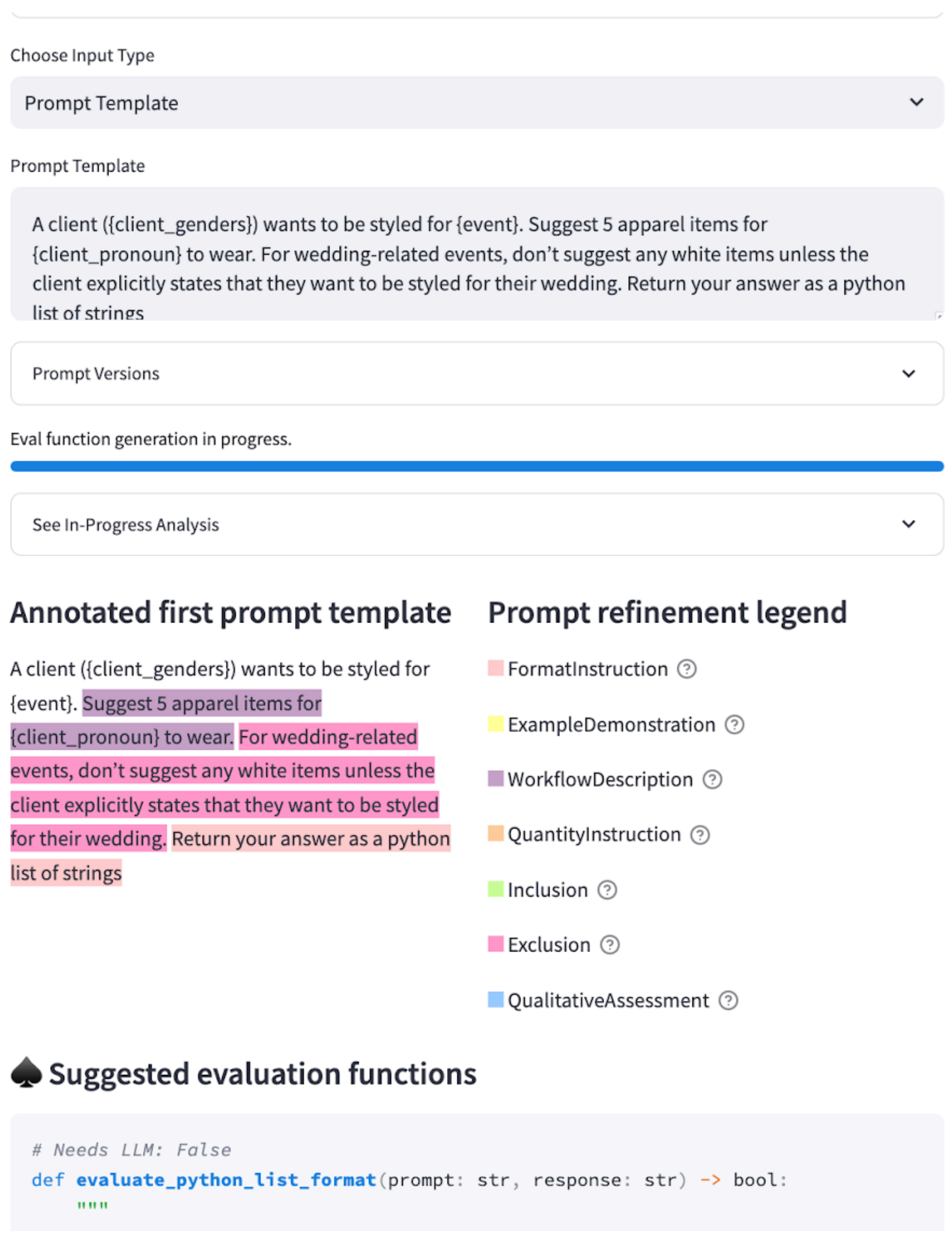
Code-based & LLM-based implementations

Shankar, Shreya, Haotian Li, Parth Asawa, Madelon Hulsebos, Yiming Lin, J. D. Zamfirescu-Pereira, Harrison Chase, Will Fu-Hinthorn, Aditya G. Parameswaran, and Eugene Wu. "SPADE: Synthesizing Data Quality Assertions for Large Language Model Pipelines." *Under submission.*

Lessons Learned From Large-Scale Deployment

- Deployed a version with LangChain in November 2023
- Findings across 2000+ LLM pipelines
  - Inclusion & exclusion assertions were most common
  - Redundant assertions
  - Incorrect assertions





Shankar, Shreya, Haotian Li, Parth Asawa, Madelon Hulsebos, Yiming Lin, J. D. Zamfirescu-Pereira, Harrison Chase, Will Fu-Hinthorn, Aditya G. Parameswaran, and Eugene Wu. "SPADE: Synthesizing Data Quality Assertions for Large Language Model Pipelines." *Under submission.*

Problems with Candidate Assertions

- Redundancy

```
def assert_sensitive_1(prompt, response):
  return “race” not in response and
  “gender” not in response and “name” not
  in response
```

```
def assert_sensitive_3(prompt, response):
  return ask_llm(f”Is there sensitive
  information like race or gender in
  {response}?”)
```

- Incorrectness

“Shreya Shankar, an Indian-American female…”

✅ `assert_sensitive_1`

- Why not eyeball and deduplicate?
    - 50+ assertions for 5+ prompt versions
    - Don’t know ask_llm accuracy

Shankar, Shreya, Haotian Li, Parth Asawa, Madelon Hulsebos, Yiming Lin, J. D. Zamfirescu-Pereira, Harrison Chase, Will Fu-Hinthorn, Aditya G. Parameswaran, and Eugene Wu. "SPADE: Synthesizing Data Quality Assertions for Large Language Model Pipelines." *Under submission.*

Filtering Candidate Assertions

Given all candidate assertions and user-provided grades on LLM pipeline outputs, select a minimal set of assertions, subject to constraints on:

- Coverage of failures

  fraction of bad outputs flagged by at least one selected assertion

- False failure rate (accuracy)

  fraction of good outputs flagged by at least one selected assertion

Can formulate as an ILP

Shankar, Shreya, Haotian Li, Parth Asawa, Madelon Hulsebos, Yiming Lin, J. D. Zamfirescu-Pereira, Harrison Chase, Will Fu-Hinthorn, Aditya G. Parameswaran, and Eugene Wu. "SPADE: Synthesizing Data Quality Assertions for Large Language Model Pipelines." *Under submission.*

Filtering Candidate Assertions

What happens if user-provided grades don't encompass all failure modes?

- Select a minimal set of assertions, subject to constraints on *coverage* and *false failure rate*
- Solution is hyper-specific to user-provided grades
  - May drop useful assertions, e.g.,

```
def assert_tone(prompt, response):
  return ask_llm(f"Is the tone here professional: {response}?")
```

→ If this passes for all graded outputs ✅ ✅ ✅ ✅ ✅ ✅ ✅…it gets filtered out by the optimizer!

- Can't expect people to provide exhaustive graded samples

Shankar, Shreya, Haotian Li, Parth Asawa, Madelon Hulsebos, Yiming Lin, J. D. Zamfirescu-Pereira, Harrison Chase, Will Fu-Hinthorn, Aditya G. Parameswaran, and Eugene Wu. "SPADE: Synthesizing Data Quality Assertions for Large Language Model Pipelines." *Under submission.*

Filtering Candidate Assertions

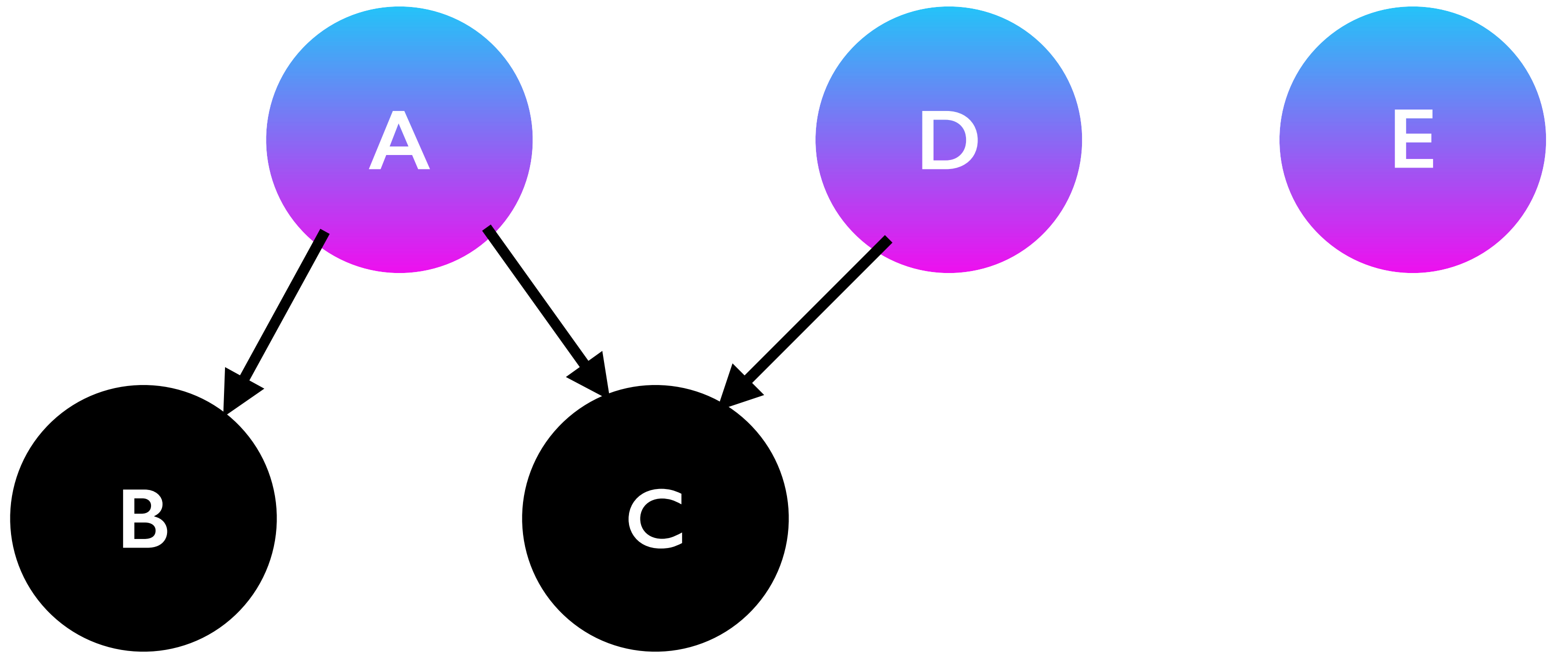
With an incomplete graded sample of LLM outputs

Idea: derive a *subsumption graph* and incorporate this into the ILP

```
def assert_num_items(prompt, response):
  # try to load into JSON object
  # check that there are > 5 items
  ...
```

⟹

```
def assert_json(prompt, response):
  # try to load into a JSON object
  ...
```



Penalize objective if these nodes

are **not** included in the solution

Shankar, Shreya, Haotian Li, Parth Asawa, Madelon Hulsebos, Yiming Lin, J. D. Zamfirescu-Pereira, Harrison Chase, Will Fu-Hinthorn, Aditya G. Parameswaran, and Eugene Wu. "SPADE: Synthesizing Data Quality Assertions for Large Language Model Pipelines." *Under submission.*

SPADE Empirical Study

- 9 LLM pipelines across various fields (coding, finance, education)
- Subsumption-based solution outperforms when grading doesn't cover all failure modes
- Baseline selecting individual assertions meeting the FFR threshold fails in aggregate, e.g.,
  - `assertion_one` FFR = 10%
  - `assertion_two` FFR = 15%
  - `assertion_one & assertion_two` FFR $\leq$ 25%
  - Takeaway: evaluation assistants must consider *interactions between* assertions

Pipeline	# CA	Method	FFR	Coverage on E'	Frac Func. Selected	Frac Excl. Funcs. not Subsumed
codereviews	44	BASELINE	0.117 ✓	1 ✓	0.456 (20)	0 (0)
		$\text{SPADE}_{\text{cov}}$	0 ✓	0.625 ✓	0.045 (2)	0.409 (18)
		$\text{SPADE}_{\text{sub}}$	0.117 ✓	0.875 ✓	0.341 (15)	0 (0)
emails	24	BASELINE	0 ✓	1 ✓	0.5 (12)	0 (0)
		$\text{SPADE}_{\text{cov}}$	0 ✓	1 ✓	0.0417 (1)	0.458 (11)
		$\text{SPADE}_{\text{sub}}$	0 ✓	1 ✓	0.458 (11)	0 (0)
fashion	106	BASELINE	0.878 ✗	0.971 ✓	0.632 (67)	0 (0)
		$\text{SPADE}_{\text{cov}}$	0.245 ✓	0.6 ✓	0.028 (3)	0.321 (34)
		$\text{SPADE}_{\text{sub}}$	0.224 ✓	0.62 ✓	0.377 (40)	0 (0)
finance	47	BASELINE	0.667 ✗	1 ✓	0.787 (37)	0 (0)
		$\text{SPADE}_{\text{cov}}$	0.229 ✓	0.673 ✓	0.085 (4)	0.553 (26)
		$\text{SPADE}_{\text{sub}}$	0.208 ✓	0.981 ✓	0.553 (26)	0 (0)
lecturesum.	70	BASELINE	0.528 ✗	1 ✓	0.457 (32)	0 (0)
		$\text{SPADE}_{\text{cov}}$	0.194 ✓	0.643 ✓	0.014 (1)	0.414 (29)
		$\text{SPADE}_{\text{sub}}$	0.194 ✓	1 ✓	0.343 (24)	0 (0)
negotiation	50	BASELINE	0.444 ✗	1 ✓	0.4 (20)	0 (0)
		$\text{SPADE}_{\text{cov}}$	0.222 ✓	0.632 ✓	0.04 (2)	0.32 (16)
		$\text{SPADE}_{\text{sub}}$	0.185 ✓	1 ✓	0.34 (17)	0 (0)
sportroutine	26	BASELINE	0.211 ✓	1 ✓	0.538 (14)	0 (0)
		$\text{SPADE}_{\text{cov}}$	0.211 ✓	0.774 ✓	0.077 (2)	0.462 (12)
		$\text{SPADE}_{\text{sub}}$	0 ✓	0.871 ✓	0.308 (8)	0 (0)
statsbot	15	BASELINE	0 ✓	1 ✓	0.467 (7)	0 (0)
		$\text{SPADE}_{\text{cov}}$	0 ✓	0.935 ✓	0.133 (2)	0.333 (5)
		$\text{SPADE}_{\text{sub}}$	0 ✓	1 ✓	0.467 (7)	0 (0)
threads	34	BASELINE	0 ✓	1 ✓	0.765 (26)	0 (0)
		$\text{SPADE}_{\text{cov}}$	0 ✓	0.875 ✓	0.029 (1)	0.735 (25)
		$\text{SPADE}_{\text{sub}}$	0 ✓	1 ✓	0.589 (20)	0 (0)

Table 4: Results of different versions of SPADE with $\alpha = 0.6$ and $\tau = 0.25$. "# CA" is short for the number of candidate assertions. The ✓and ✗ marks denote whether α and τ constraints are met. Each entry is a fraction of the total number of candidate assertions for that pipeline (with the absolute number in parentheses). $\text{SPADE}_{\text{cov}}$ selects the fewest assertions overall. $\text{SPADE}_{\text{sub}}$ selects the fewest assertions while optimizing for subsumption.

Shankar, Shreya, Haotian Li, Parth Asawa, Madelon Hulsebos, Yiming Lin, J. D. Zamfirescu-Pereira, Harrison Chase, Will Fu-Hinthorn, Aditya G. Parameswaran, and Eugene Wu. "SPADE: Synthesizing Data Quality Assertions for Large Language Model Pipelines." *Under submission.*

Evaluation Assistants

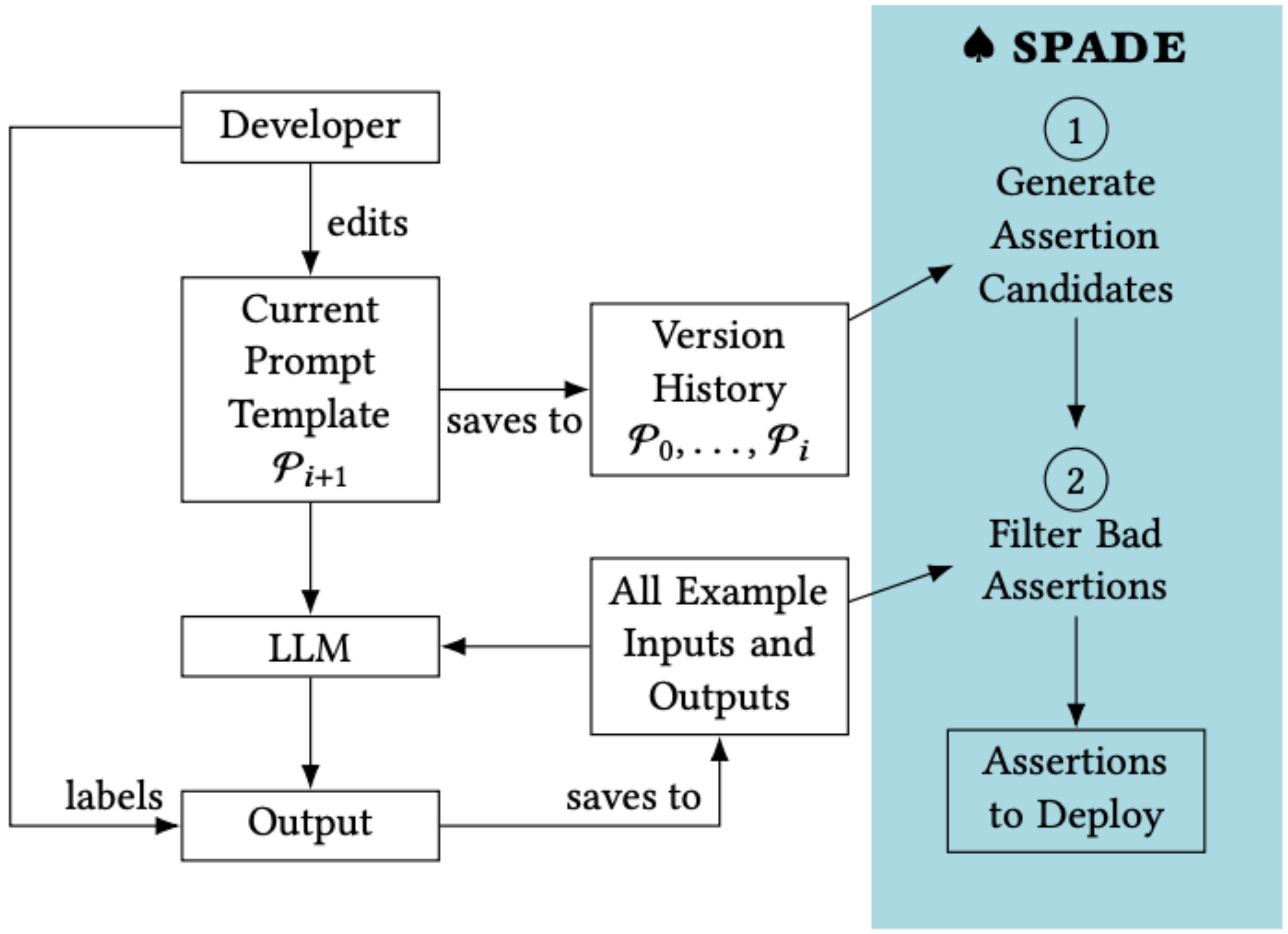
- *Evaluation assistants:* tools that aid humans in creating evaluations and assertions that *align* with how they would grade pipeline outputs
- Today's talk:
  - Auto-generating criteria and assertions
  - Insights from large-scale deployment with LangChain
  - Mixed-initiative interface to develop custom assertions
  - Lessons learned from small-scale qualitative study

Shankar, Shreya, Haotian Li, Parth Asawa, Madelon Hulsebos, Yiming Lin, J. D. Zamfirescu-Pereira, Harrison Chase, Will Fu-Hinthorn, Aditya G. Parameswaran, and Eugene Wu. "SPADE: Synthesizing Data Quality Assertions for Large Language Model Pipelines." *Under submission.*

Shankar, Shreya, J. D. Zamfirescu-Pereira, Bjoern Hartmann, Aditya G. Parameswaran, and Ian Arawjo. "Who Validates The Validators? Aligning LLM-Assisted Evaluation of LLM Outputs with Human Preferences." *Under submission.*

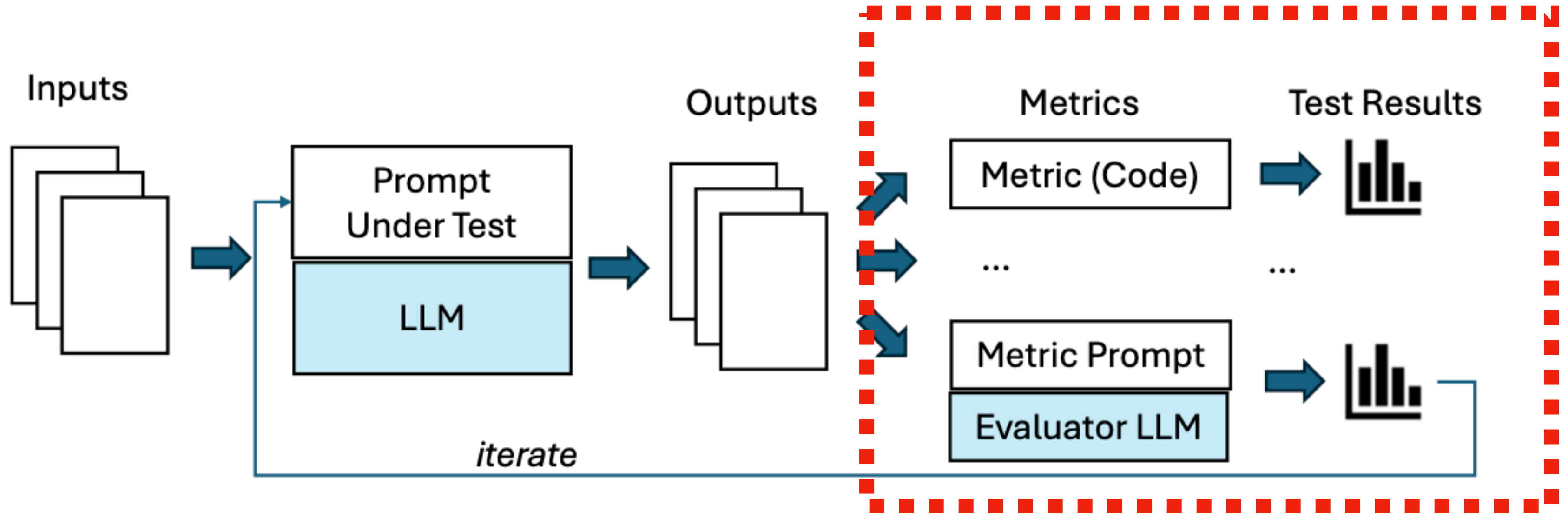
Incorporating Humans Into the Workflow

- SPADE takes a *long* time to execute
  - Need grades upfront
  - LLM latencies (minutes!)
  - Resulting assertions still might not be perfect, requiring iteration & human input
- How can we design an interface to (1) *support rapid iteration* while (2) *maintaining or improving assertion alignment with human expectations?*



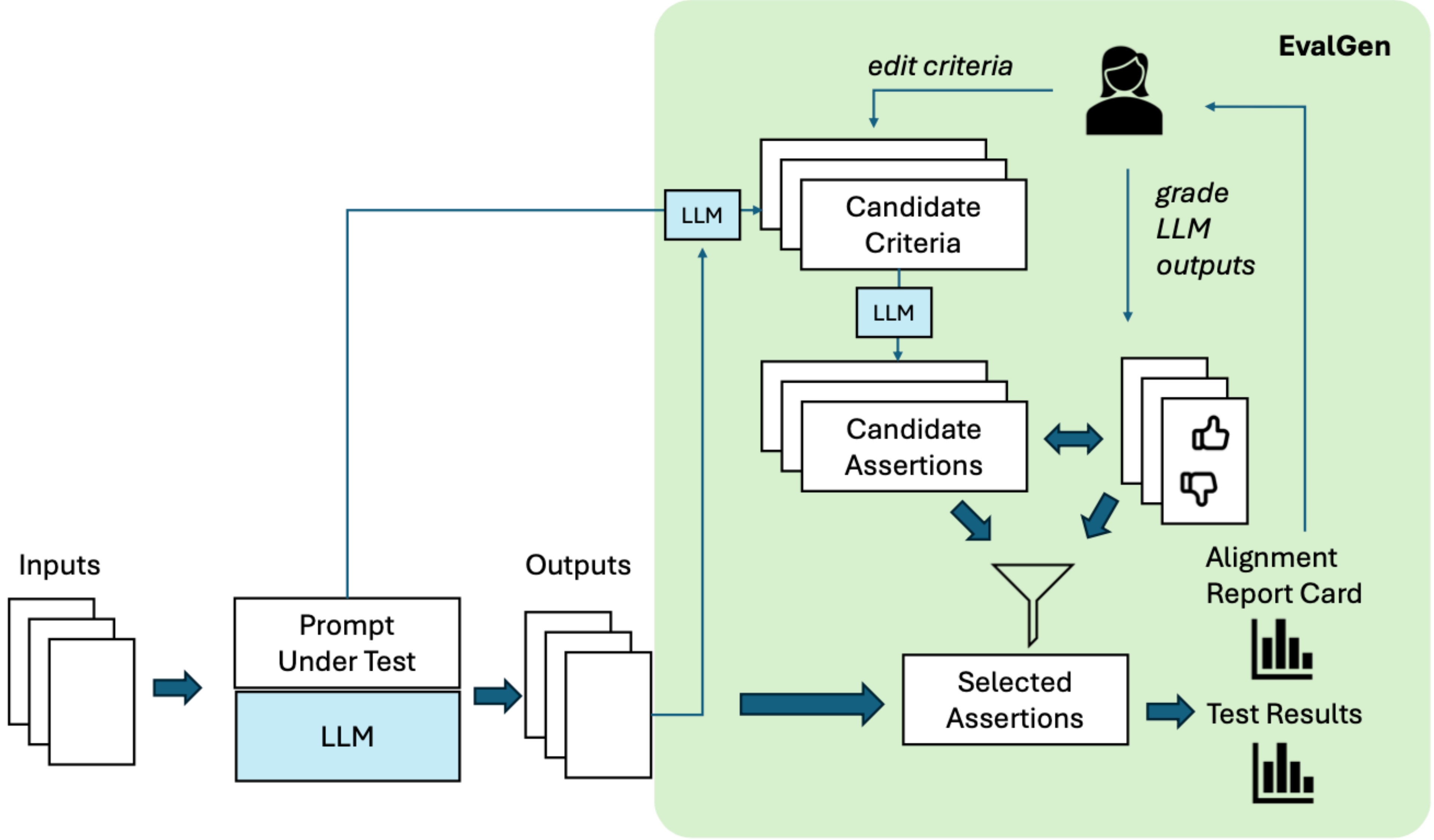
Interfaces for Evaluation Assistants

- To support iteration, we need to minimize wait time
- Can solicit human input *throughout* the assertion generation, filtering, and assessment workflows
    - Humans can edit criteria
    - Humans can grade LLM outputs



Lots of wait time here

(a) **Typical Evaluation Pipeline**



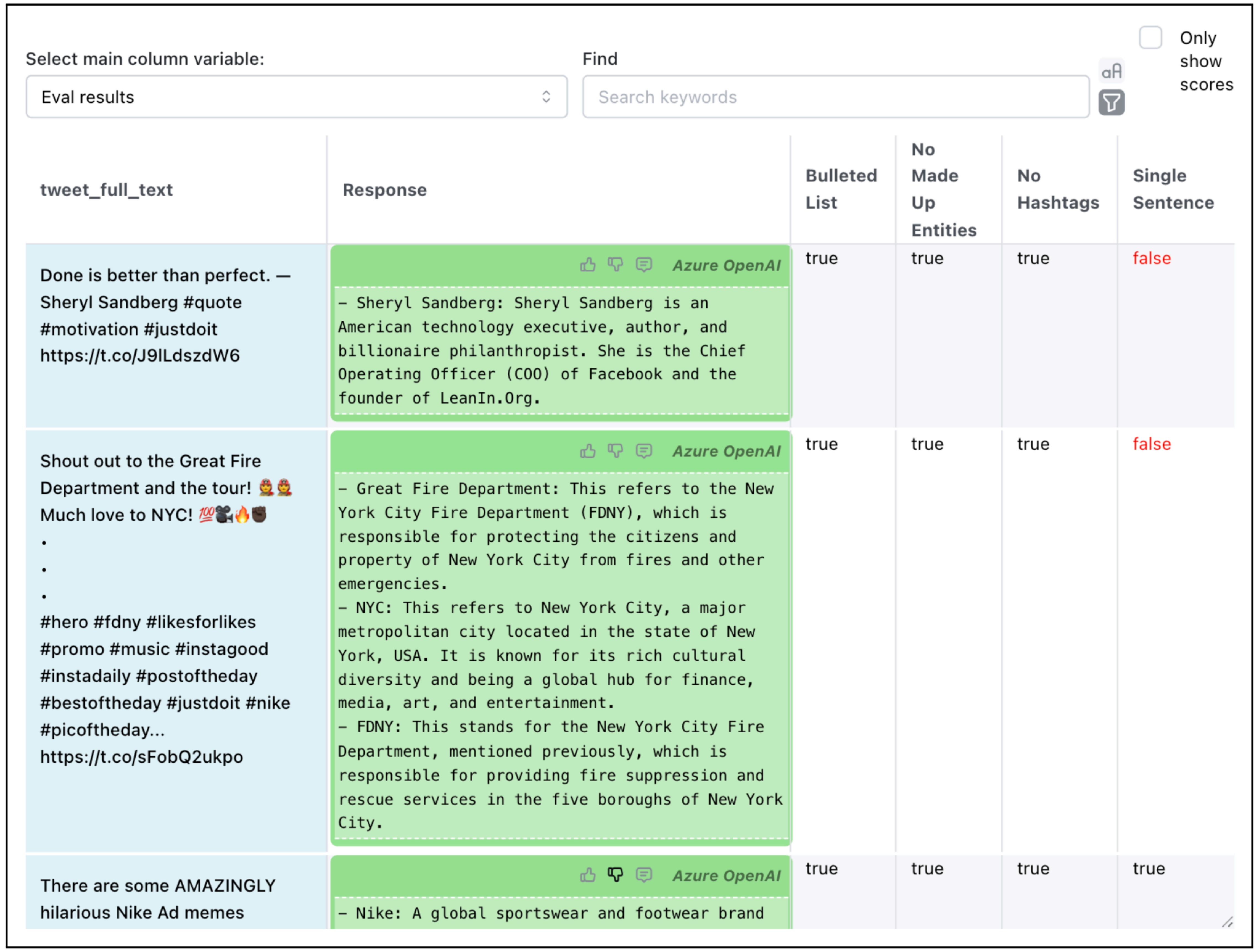
(b) **The EvalGen Evaluation Pipeline**

Shankar, Shreya, J. D. Zamfirescu-Pereira, Bjoern Hartmann, Aditya G. Parameswaran, and Ian Arawjo. "Who Validates The Validators? Aligning LLM-Assisted Evaluation of LLM Outputs with Human Preferences." *Under submission.*

EvalGen Interface

Assertion generation & alignment via a sample

Scaling up to *all* (ungraded) outputs



Shankar, Shreya, J. D. Zamfirescu-Pereira, Bjoern Hartmann, Aditya G. Parameswaran, and Ian Arawjo. "Who Validates The Validators? Aligning LLM-Assisted Evaluation of LLM Outputs with Human Preferences." *Under submission.*

Qualitative Study: *How do people use EvalGen?*

- 60-minute studies with 9 ML and AI engineers in industry who had prior experience building LLM pipelines
- We asked participants to use EvalGen in an *open-ended way* to come up with assertions for an LLM pipeline: either their own pipeline or our example pipeline (named entity recognition/ NER on tweets)
- Participants liked EvalGen as a *starting point* for assertions
- Participants had mixed opinions on assertion alignment

P1	P2	P3	P4	P5	P6	P7	P8	P9
6	5	3	4	5	3	1	2	5

Table 2: Ratings (1-7, 7 best) for the statement, "*I felt like the assertions aligned with my grades.*" Responses were mixed.

Shankar, Shreya, J. D. Zamfirescu-Pereira, Bjoern Hartmann, Aditya G. Parameswaran, and Ian Arawjo. "Who Validates The Validators? Aligning LLM-Assisted Evaluation of LLM Outputs with Human Preferences." *Under submission.*

Criteria Drift

Why is assertion alignment/trust so hard to achieve?

- Grading LLM outputs spurred changes or refinements to evaluation criteria
  - Adding new criteria
  - Reinterpret criteria to better fit the LLM's behavior
- Sensemaking is a part of grading
- Implications: grading must be a *continual* process, as prompts, LLMs, and pipelines change

Extract all entities from this tweet: {input}. Don't extract hashtags as entities.

Criteria: *no hashtags as entities*

Input Tweet	Input Tweet	Input Tweet
… #JustDoIt	*… #ColinKaepernick*	*… #Nike*
LLM Output	**LLM Output**	**LLM Output**
** #justdoit: The Nike slogan…*	** Colin Kaepernick: former football…*	** Nike: a shoe company…*
❌	✅	✅
	"Hm I said no hashtags as entities but I think the LLM did the right thing here"	*"I'm failing everything…I think actually the criteria should just be no # in the output"*

Shankar, Shreya, J. D. Zamfirescu-Pereira, Bjoern Hartmann, Aditya G. Parameswaran, and Ian Arawjo. "Who Validates The Validators? Aligning LLM-Assisted Evaluation of LLM Outputs with Human Preferences." *Under submission.*

Code-Based Evals != LLM-Based Evals

Why is assertion alignment/trust so hard to achieve?

- Grading outputs is good to align LLM-based evals, not code-based evals
  - *"When something can be solved using Python code, I do have an envisioned [implementation] in mind that I can easily verify. Just showing [me] the [code] will be quicker."*
- Use LLM-based evaluators when criteria is "fuzzy" or when input data is dirty

Input Tweet

… #Kapernick

LLM Output

** Kaepernick: former football…*

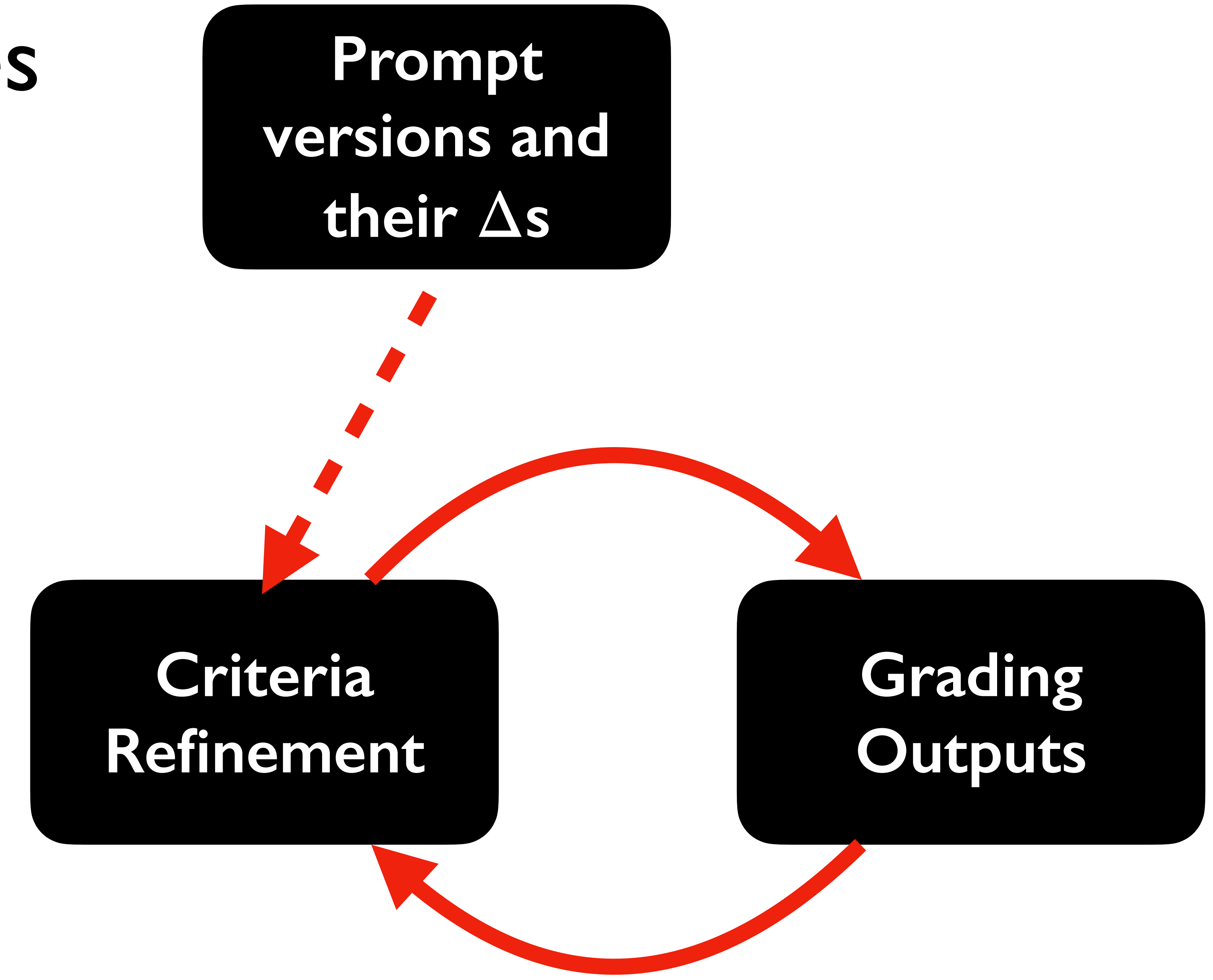
assert entity_name in input ❌

ask_llm("Is each retrieved entity in the original input tweet?") ✅

Shankar, Shreya, J. D. Zamfirescu-Pereira, Bjoern Hartmann, Aditya G. Parameswaran, and Ian Arawjo. "Who Validates The Validators? Aligning LLM-Assisted Evaluation of LLM Outputs with Human Preferences." *Under submission.*

Evaluation Assistants: Overall Takeaways

- When running LLMs at scale…there will be mistakes
- Prompt deltas can inform assertion criteria
- **There is no "ground-truth" set of grades!**
  - Assertions need to evolve as data and LLM pipelines evolve
- Assertion generation and selection is an *iterative process* steered by humans



Shankar, Shreya, Haotian Li, Parth Asawa, Madelon Hulsebos, Yiming Lin, J. D. Zamfirescu-Pereira, Harrison Chase, Will Fu-Hinthorn, Aditya G. Parameswaran, and Eugene Wu. "SPADE: Synthesizing Data Quality Assertions for Large Language Model Pipelines." *Under submission.*

Shankar, Shreya, J. D. Zamfirescu-Pereira, Bjoern Hartmann, Aditya G. Parameswaran, and Ian Arawjo. "Who Validates The Validators? Aligning LLM-Assisted Evaluation of LLM Outputs with Human Preferences." *Under submission.*