

APAC 2026

An Agentic Approach to Validation, Influencing Reporting with AI

Junibel R. De Leon, Pfizer
Karthik Palalini, Pfizer

Table 14.3.1.2.1 Adverse Events by MedDRA PT - Safety Set

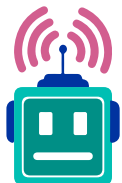
ADAE

AEDECOD

SAFFL=="Y"



Time Spent: 15-30 mins



Time Saved: ~20 mins



Grab some chai!



Disclaimer

We confirm that, the opinions and thoughts discussed in this presentation are subject to our own independent views and are not subject to the opinions of the organization that we represent.

Today's Agenda

Objective

Workflow

Models Evaluated

Results

Strengths and Limitations

Future Work



Objective

Can an Agentic AI follow QC logic?



Interpret table logic



Retrieve step instructions



Generate R code



Enable human oversight



Current Scope: Simplified AE summary by PT



Workflow



Mockshell

USER PROMPT



ADaM
Data

RAG



MODEL-CODE
GENERATION

POST-
PROCESSING



R code
for Summary Table

report_name	step	instruction
summary of adverse event by meddra PT or preferred term	1	Filter ADSL
summary of adverse event by meddra PT or preferred term	2	Count TRTVAR in ADSL
summary of adverse event by meddra PT or preferred term	3	Rename count variable n to N
summary of adverse event by meddra PT or preferred term	4	Join ADSL denominator to ADAE by TRTVAR
summary of adverse event by meddra PT or preferred term	5	Filter ADAE dataset
summary of adverse event by meddra PT or preferred term	6	Group data by TRTVAR
summary of adverse event by meddra PT or preferred term	7	Count adverse event by TRTVAR and AEDECOD
summary of adverse event by meddra PT or preferred term	8	Calculate percentage using denominator
summary of adverse event by meddra PT or preferred term	9	Create display value as "count (percent)"
summary of adverse event by meddra PT or preferred term	10	Set row label as AE preferred term
summary of adverse event by meddra PT or preferred term	11	Ungroup data
summary of adverse event by meddra PT or preferred term	12	Rename treatment column to _TRT
summary of adverse event by meddra PT or preferred term	13	Keep only row label, value and treatment
summary of adverse event by meddra PT or preferred term	14	Sort output by treatment and row label
summary of adverse event by meddra PT or preferred term	15	Compare the production dataset and validation dataset

```
===== GENERATED QC PROGRAM =====

# Step 1
adsl |> filter(adsl_filter)

# Step 2
adsl |> count(TRTVAR)

# Step 3
adsl |> rename(N = n)

# Step 4
data |> left_join(adsl, by = "TRTVAR")

# Step 5
data |> filter(adae_filter)

# Step 6
data |> count(TRTVAR, AEDECOD, N)
```

```
# -----
# User-defined filters
# -----
adsl_filter <- ARMCD != ""
adae_filter <- SAFFL == "Y" & TRTENFL == "Y"

# ADSL: Treatment counts (denominator)
# -----
adsl_qc <- adsl %>%
  filter(adsl_filter) %>%
  count(ARMCD) %>%
  rename(N = n)

# -----
# ADAE: Detail-level summary
# -----
data1 <- adae %>%
  left_join(adsl, by = "ARMCD") %>%
  filter(adae_filter) %>%
  count(ARMCD, AEDECOD, N) %>%
  group_by(ARMCD) %>%
  mutate(percent = round(100 * n / N, 1)) %>%
  mutate(_CVALUE = paste0(n, " (", percent, "%)") %>%
  mutate(_RHLABEL = AEDECOD) %>%
  ungroup() %>%
  rename(_TRT = ARMCD) %>%
  select(_RHLABEL, _CVALUE, _TRT) %>%
  arrange(_TRT, _RHLABEL)
```

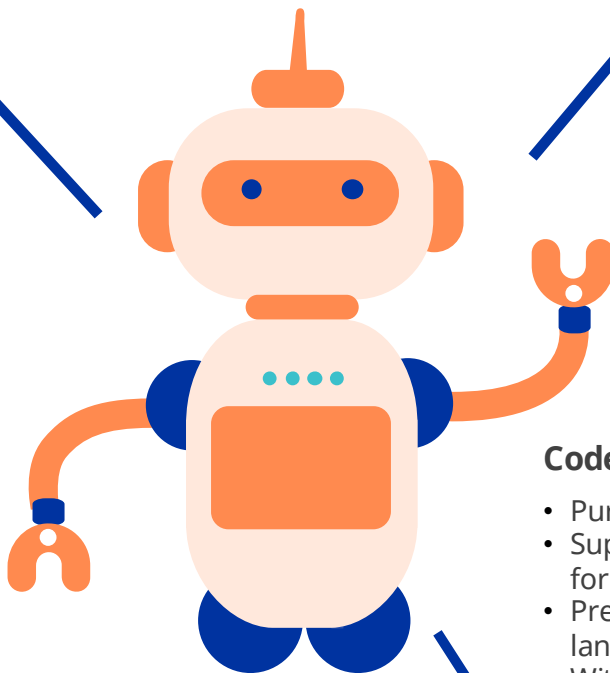
user_prompt = "do the qc validation of summary of adverse event by preferred term"



Models Evaluated

Flan-T5 base

- Text-to-text model optimized for summarization/ translation/ NLU, note code generation
- Lower text-to-code performance vs a code-specialized model



DeepSeek

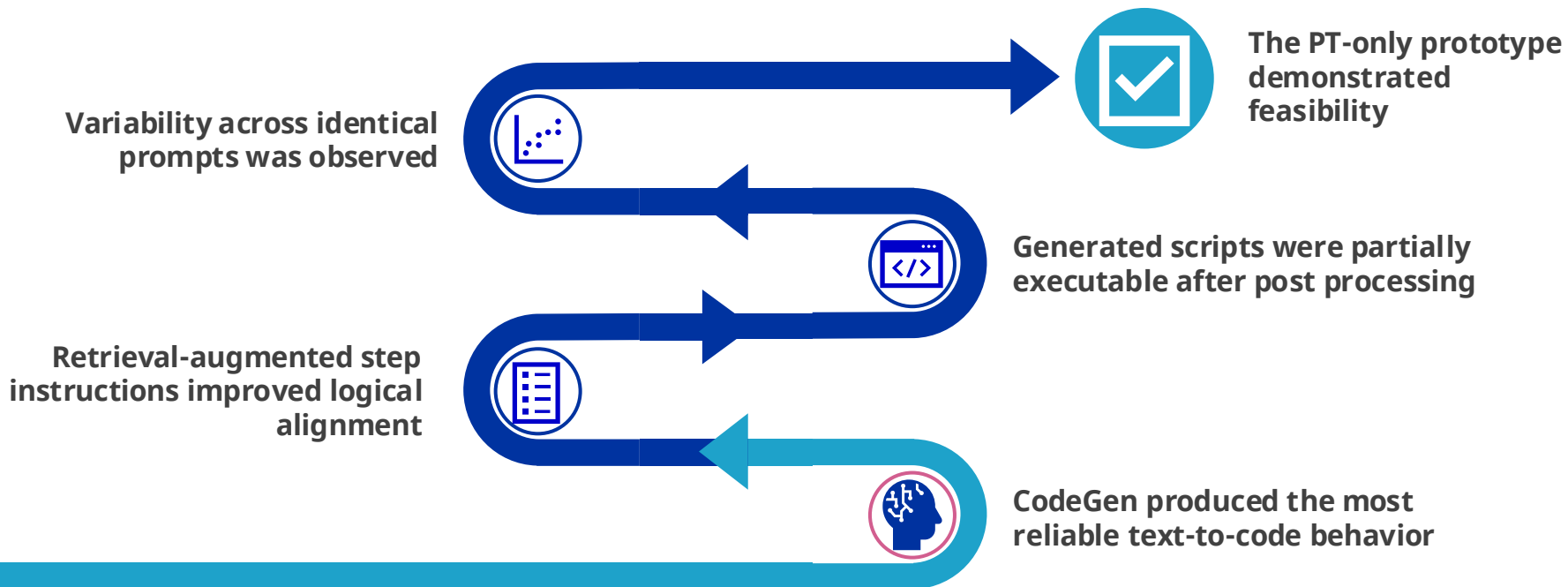
- Lightweight, efficient code model for fast inference in resource-constrained environments
- Best for code completion, bug fixing
- Reasonably accurate without fine-tuning; further training not pursued due to time/compute constraints

CodeGen

- Purpose-built for code generation
- Supports interactive, multi-turn refinement for complex tasks
- Pretrained on multiple programming languages
- With focused R fine-tuning, reduces hallucinations and improves functional correctness



Results





Strengths



Logical step sequences



Clear starting point



Transparent process

Limitations



Output inconsistency



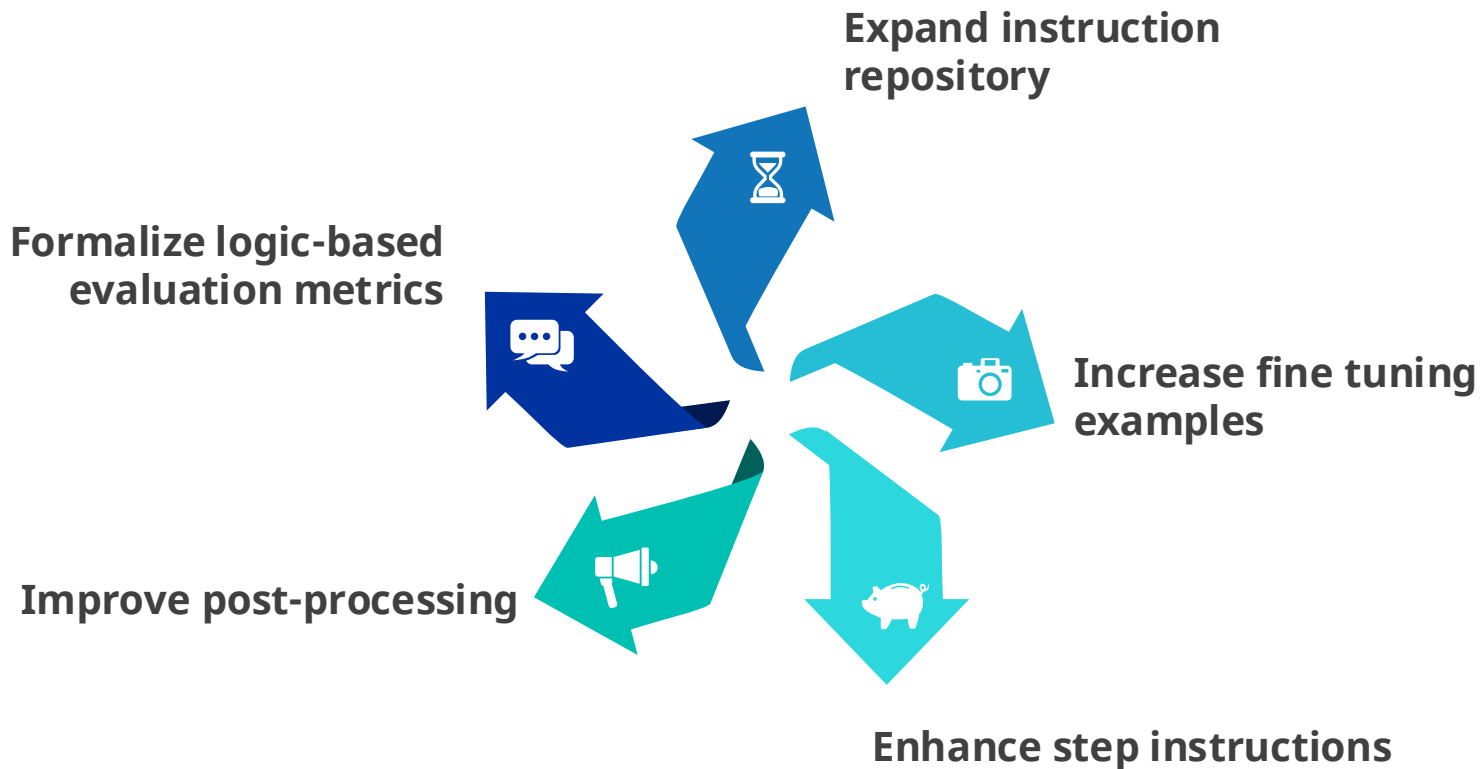
Syntax fixes needed



Limited training data



PT-only scope





Summary



**Agentic AI can begin
to follow QC
programmer logic**



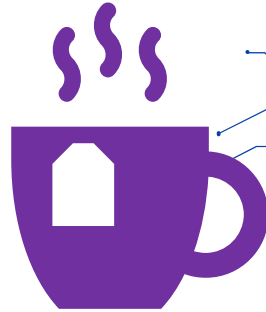
**CodeGen produced
the strongest text-to-
code performance**



**Early prototype, clear
direction forward**



Final Thoughts





Final Thoughts





Acknowledgement

- ❖ Muhammed Farzan K
- ❖ Mayur Mishra
- ❖ Alisha D
- ❖ Periasamy Karuppannan
- ❖ Priya Venkatesan Iyer
- ❖ Debjit Biswas

THANK YOU



**The Global Healthcare
Data Science Community**

Contact Channels

📞 UK +44 1843 609600
✉ office@phuse.global
🌐 phuse.global

Social Media

🐦 [@phusetwitta](https://twitter.com/phusetwitta)
📘 [/phusebook](https://facebook.com/phusebook)
▶ [/phusetube](https://youtube.com/phusetube)
🌐 [/company/phuse](https://linkedin.com/company/phuse)