

Module 8:

Prompt Engineering & LLMs

Mastering predictability with gpt-4o-mini and structured outputs.

Foundational **Mechanics**

Understanding the core engine behind Large Language Models.

The Transformer Engine

Attention Mechanism

LLMs don't read text; they process **contextual relationships**. The attention mechanism allows the model to weigh the importance of different words in a sequence simultaneously.

Next-Token Prediction

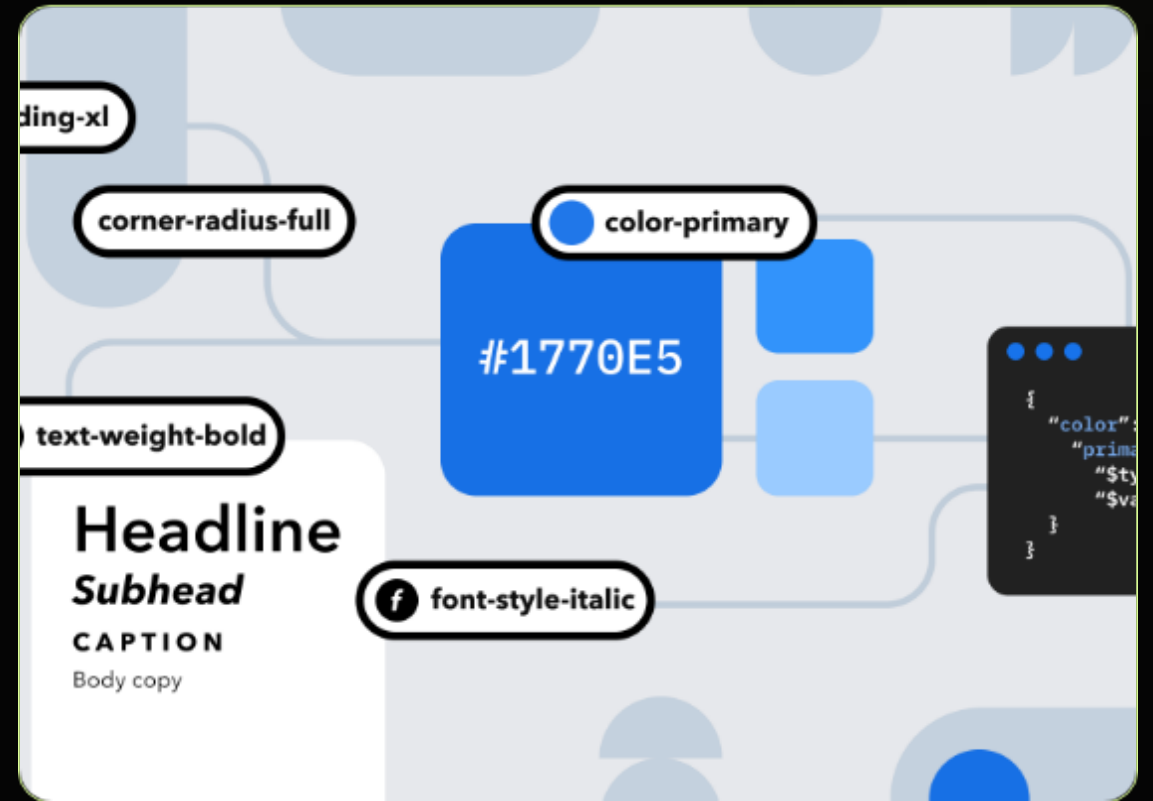
At its core, an LLM is a **probabilistic engine**. It calculates the likelihood of the next token based on all preceding data in the context window.

$$P(w_t | w_1, \dots, w_{t-1})$$

Tokenization & Encoding

Before processing, raw text is fragmented into **tokens**—the building blocks of LLM language. These are often sub-words or character chunks.

- ✓ **Efficient Storage:** Tokens reduce memory overhead.
- ✓ **Vocabulary Limit:** Models use fixed vocab sizes.
- ✓ **Numerical Mapping:** Tokens map to vector IDs.



Interaction Architecture

System vs. User roles and conversational framing.

Defining Operational Roles



System Role

The "Base Constitution". Sets global instructions, persona, and strict constraints (e.g., "Return only valid JSON").



User Role

The "Dynamic Input". Contains the variable data or specific request to be processed under system rules.



Assistant Role

The "Resultant Output". The model's response, which maintains continuity by referencing the prior history.

The Temperature Variable

0.2

DETERMINISTIC TARGET

Why 0.2 for Pipelines?

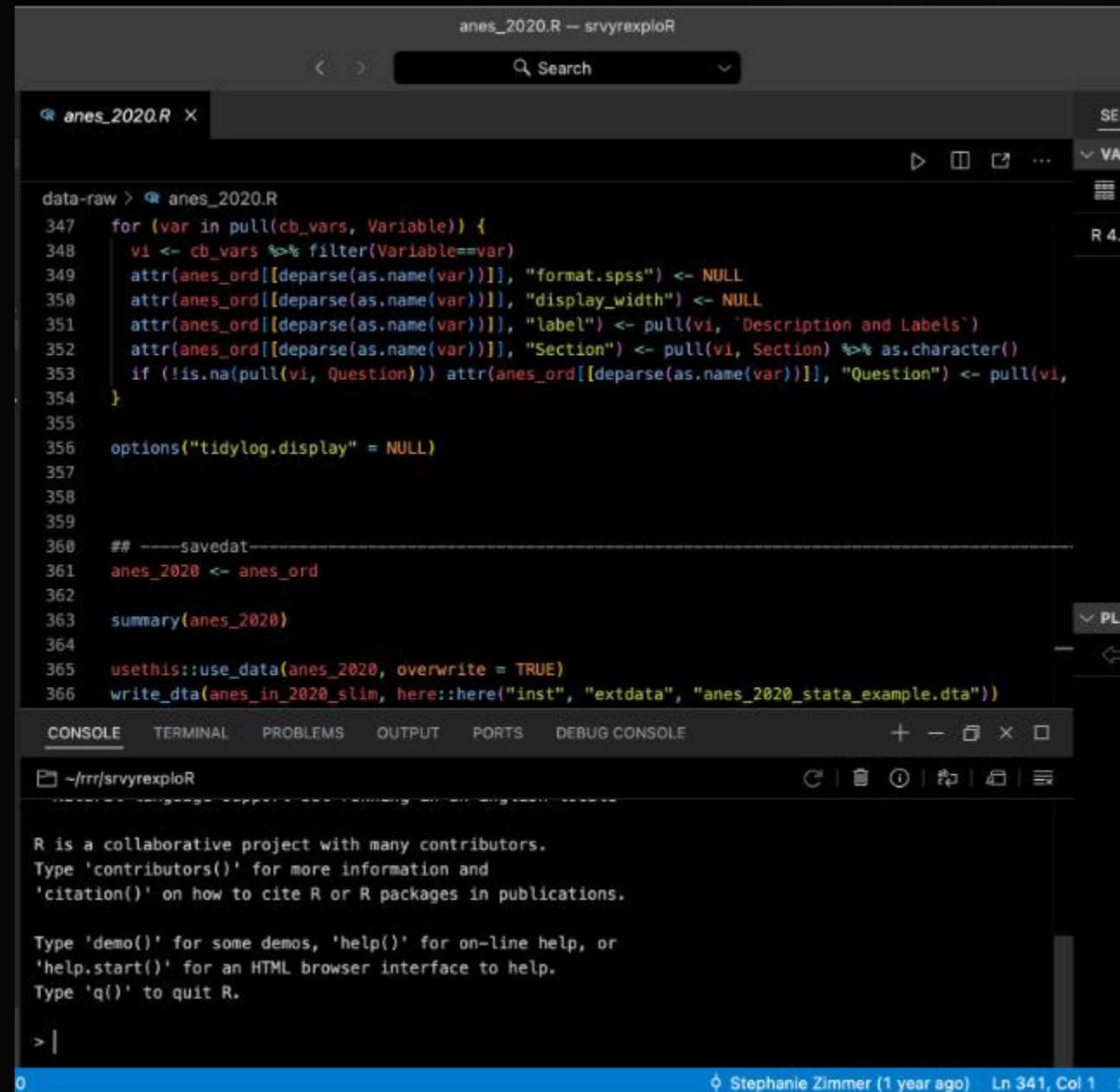
Temperature controls the randomness of the next-token selection. While 1.0 is creative, **0.2 ensures predictability.**

- **Reliability:** Consistent outputs for the same prompt.
 - **Strict Following:** High adherence to schema rules.
 - **Logic:** Better for extracting structured data.
-

Strict JSON Output

To integrate LLMs into software pipelines, output must be **machine-readable**. Using "Zero-Tolerance" personas forces the model to bypass conversational fluff.

By defining clear keys and preventing markdown wrappers, we achieve 100% syntactic validity for automated parsing.



The screenshot displays the RStudio IDE interface. The top pane shows an R script titled 'anes_2020.R' with the following code:

```
anes_2020.R  
data-row > anes_2020.R  
347 for (var in pull(cb_vars, Variable)) {  
348   vi <- cb_vars %>% filter(Variable==var)  
349   attr(anes_ord[[deparse(as.name(var))]], "format.spss") <- NULL  
350   attr(anes_ord[[deparse(as.name(var))]], "display_width") <- NULL  
351   attr(anes_ord[[deparse(as.name(var))]], "label") <- pull(vi, "Description and Labels")  
352   attr(anes_ord[[deparse(as.name(var))]], "Section") <- pull(vi, Section) %>% as.character()  
353   if (!is.na(pull(vi, Question))) attr(anes_ord[[deparse(as.name(var))]], "Question") <- pull(vi,  
354 }  
355  
356 options("tidylog.display" = NULL)  
357  
358  
359  
360 ## ----savedat-----  
361 anes_2020 <- anes_ord  
362  
363 summary(anes_2020)  
364  
365 usethis::use_data(anes_2020, overwrite = TRUE)  
366 write_dta(anes_in_2020_slim, here::here("inst", "extdata", "anes_2020_stata_example.dta"))
```

The bottom pane shows the R console with the following output:

```
~/rrr/srvyexploR  
R is a collaborative project with many contributors.  
Type 'contributors()' for more information and  
'citation()' on how to cite R or R packages in publications.  
  
Type 'demo()' for some demos, 'help()' for on-line help, or  
'help.start()' for an HTML browser interface to help.  
Type 'q()' to quit R.  
  
> |
```

The status bar at the bottom indicates the user is Stephanie Zimmer (1 year ago) and the cursor is at line 341, column 1.

Configuration: gpt-4o-mini

Parameter	Recommended Value	Impact on Deliverable
model	gpt-4o-mini	Low latency and high cost-efficiency.
temperature	0.2	Maximizes predictability and schema validation.
max_tokens	500 - 1000	Prevents runaway costs and truncated JSON.
response_format	"json_object"	Enforces valid JSON syntax at the API level.

Validation Success Rates

Raw Prompting

65%

Few-Shot Template

82%

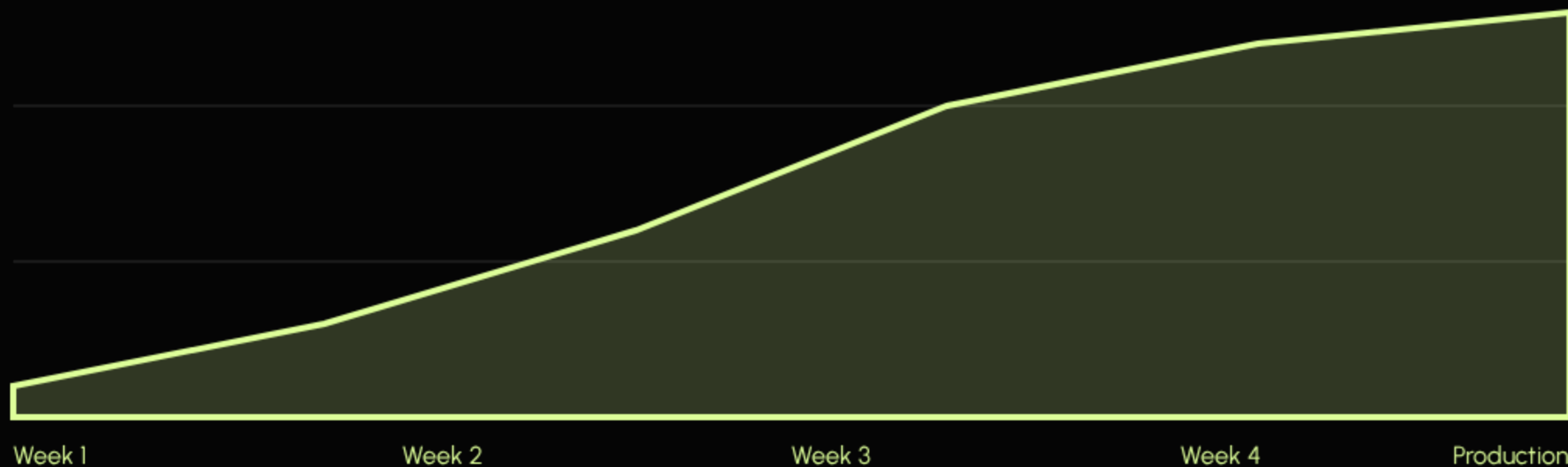
Schema-Validated Engine

98%

Implementing a prompt template engine significantly boosts predictable block generation.

Pipeline Reliability Trends

System Uptime vs. Schema Failures (Normalized)



Questions?

Next Steps: Building your custom prompt template engine.

```
ChatOpenAI(model='gpt-4o-mini', temperature=0.2)
```

Image Sources



https://images-www.contentful.com/fo9twyrwpveg/CmFq3PLtMlxCFHKRq3L2y/90b7338f38185ac08e39861acd9c7d07/STUDIO_TOKENS_AND_DESIGN_TOKENS.png

Source: www.contentful.com



https://ivelasq.rbind.io/blog/positron-theme/images/4-positron_ms_tnb.png

Source: ivelasq.rbind.io
