



ACCELERATING GROUND SCHOOL & TYPE RATINGS WITH AI COACHES

Taja Hillier - Chief Data & Artificial Intelligence Officer

OUTLINE

- *DEMO (2 examples)*
- *Workings of LLMs (High Level)*
 - ❑ *Prompt Engineering*
 - ❑ *RAG = Chunking & Embedding*
 - ❑ *Fine Tuning*
- *Workings of LLMs: Pros and Cons*

DEMO

- *DEMO 1: ATPL APP*

AI coach with Adaptive Learning Capabilities

DEMO

- *DEMO 2: API Integration*

Aquila Learning

PROMPT ENGINEERING

- *Give explicit task instructions*

Priming - Give personality	You are
Style and tone of Instructions	i.e. Acknowledge and reference received message. Thank it... Friendly, curious, professional, etc...
Handling errors and edge cases	...if you do not have information - apologise, and ask to rephrase the question or ask additional follow up questions asking for additional information to clarify the original ask.

RAG = Chunking + Embedding

- *Give explicit task instructions & Search for Relevant Content*

Priming - Give personality	You are
Style and tone of Instructions	i.e. Acknowledge and reference received message. Thank it... Friendly, curious, professional, etc...
Handling errors and edge cases	...if you do not have information - apologise, and ask to rephrase the question or ask additional follow up questions asking for additional information to clarify the original ask.
Dynamic content	Use information provided in the documentation {retrieval results}
Output Formatting	Reply as valid .json with these fields i.e. questions, answers, correct_answer, ...

RAG = Chunking + Embedding for LLMs

- **Chunking** (splitting up your document) is a crucial step that can make or break
- your model's performance.



Smaller chunks focus on specifics, sentence level



Larger chunks capture broad meanings



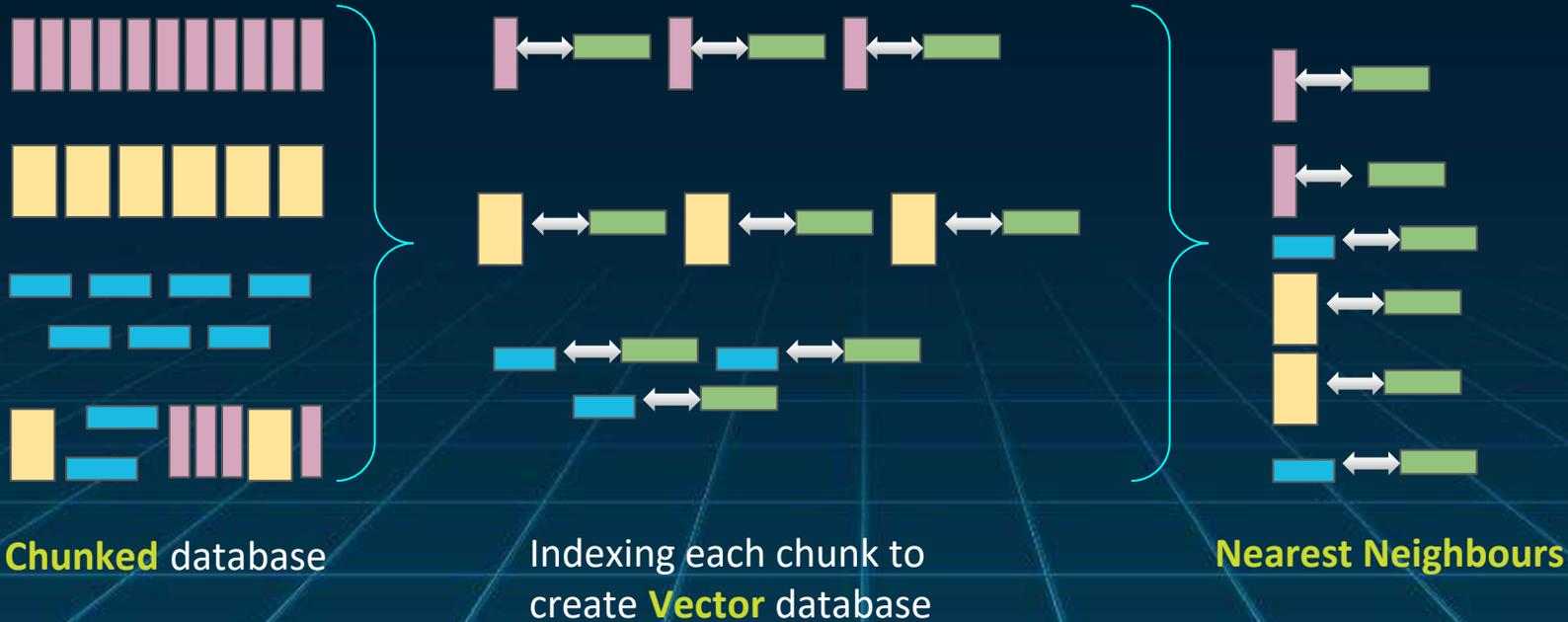
Overlapping chunks preserve context and coherence



Dynamic chunks better handle significant variations in text lengths

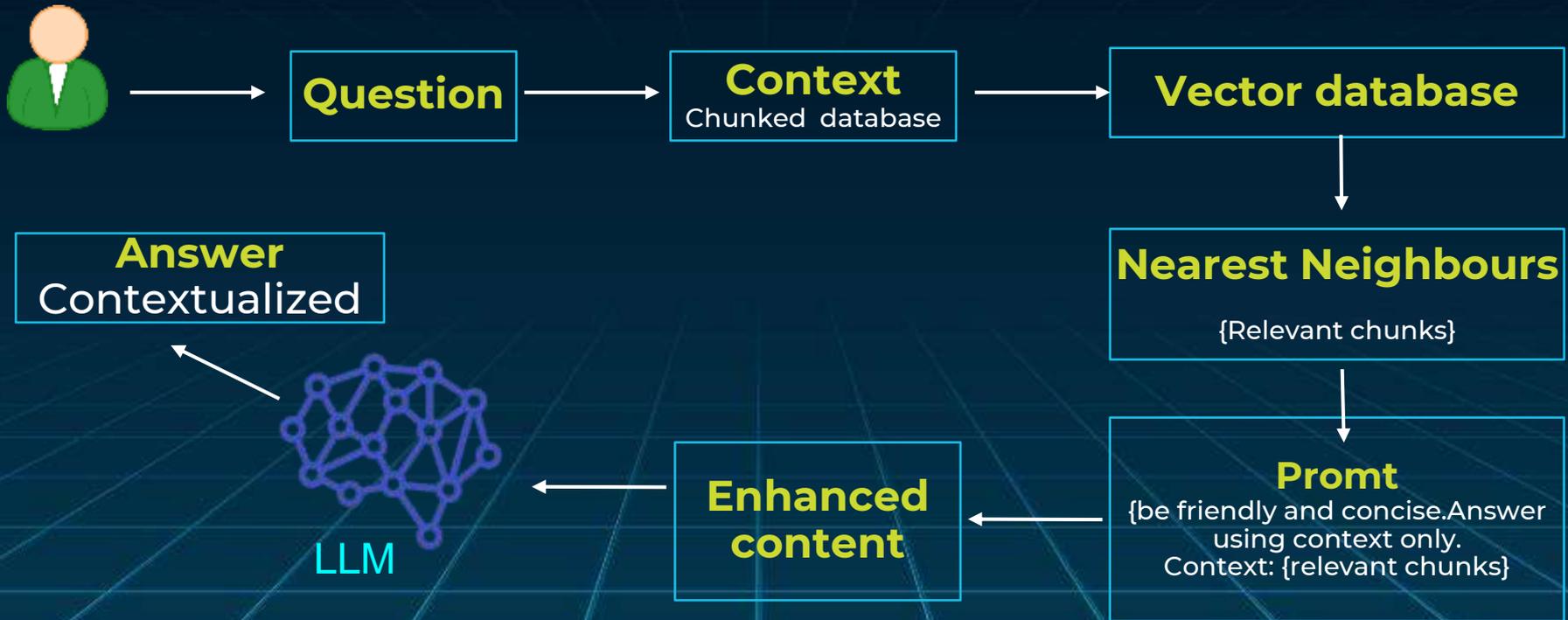
RAG + Chunking & Embedding

Embedding generation



RAG = Chunking + Embedding

- *Schematic representation or RAG Pipeline*



FINE Tuning

- *Teach model with input output examples*

When:

- For a specific task*
- Limited data*
- Optimise the cost (a trained, lower level model can perform as well as an untrained, sophisticated model)*
- Control Bias*
- Control Output*

Workings of Gen AI: PROs and CONs

Prompt Engineering	RAG	Fine tuning
<p>PROs:</p> <ul style="list-style-type: none">• No labeled data needed• Fast and intuitive prototyping	<p>PROs:</p> <ul style="list-style-type: none">• Fast iteration on prompts• Incorporates recent data without updating prompt or the model• Reduces or completely eliminates hallucinations• Output steering is more or reliable (depends on the model)	<p>PROs:</p> <ul style="list-style-type: none">• No explicit instructions needed• Faster and cheaper• Ability to steer outputs• Stability• No hallucinations
<p>CONs:</p> <ul style="list-style-type: none">• Need to handcraft a prompt• Limited by context window• Output steering less reliable	<p>CONs:</p> <ul style="list-style-type: none">• Need to handcraft a prompt• Limited by context window• Need a retrieval system (input data processing)	<p>CONs:</p> <ul style="list-style-type: none">• Need examples of how good looks like• Additional step (training)• No single way to teach the model (might require several iterations)

SUMMARY

- ***Current state of GenAI applications:***

- Prompt Engineering*
- RAG system*
- Started on fine tuning a model for one specific used case*

- **Next Steps:**

- Automate document processing i.e. chunking
- Incorporate voice, images and video formats and/or used of multimodal models
- Fine tuning

